



Performance Management for Cloud Databases via Machine Learning

Olga Papaemmanouil
Brandeis University

Outline

Motivation

Offline Learning

Online Learning

Conclusions

Outline

Motivation

Offline Learning

Online Learning

Conclusions

Cloud Databases

Challenges

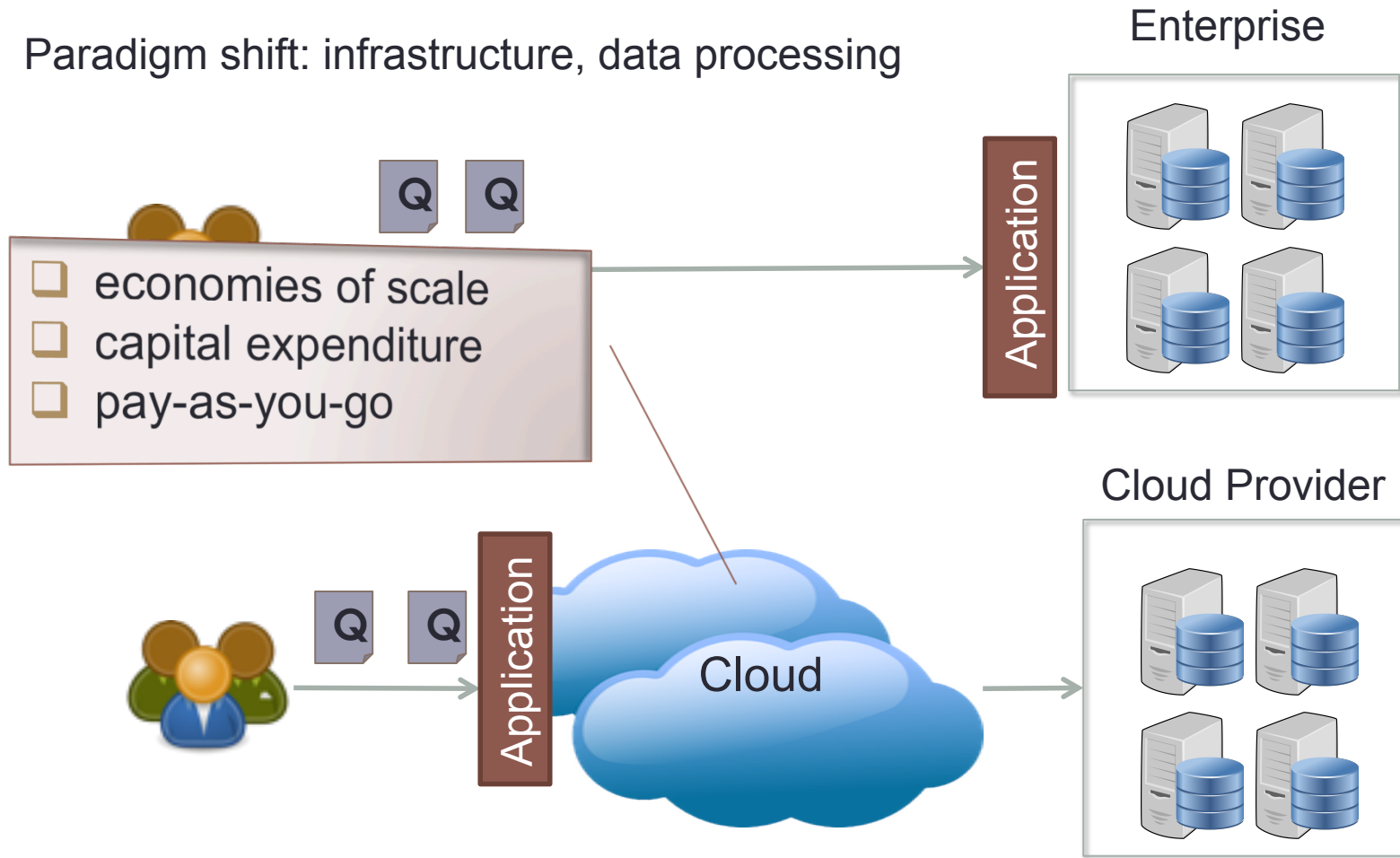
State-of-the-Art

Why Machine Learning ?

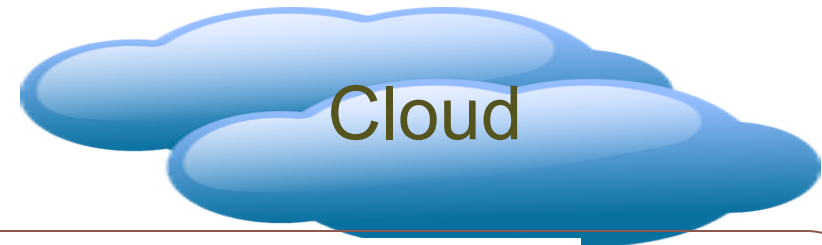
WiSeDB Advisor

Cloud Computing

Paradigm shift: infrastructure, data processing

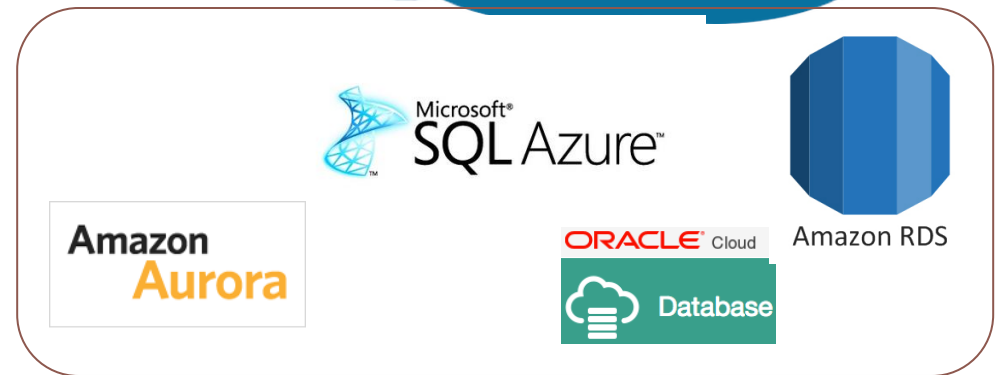


Cloud Databases Landscape



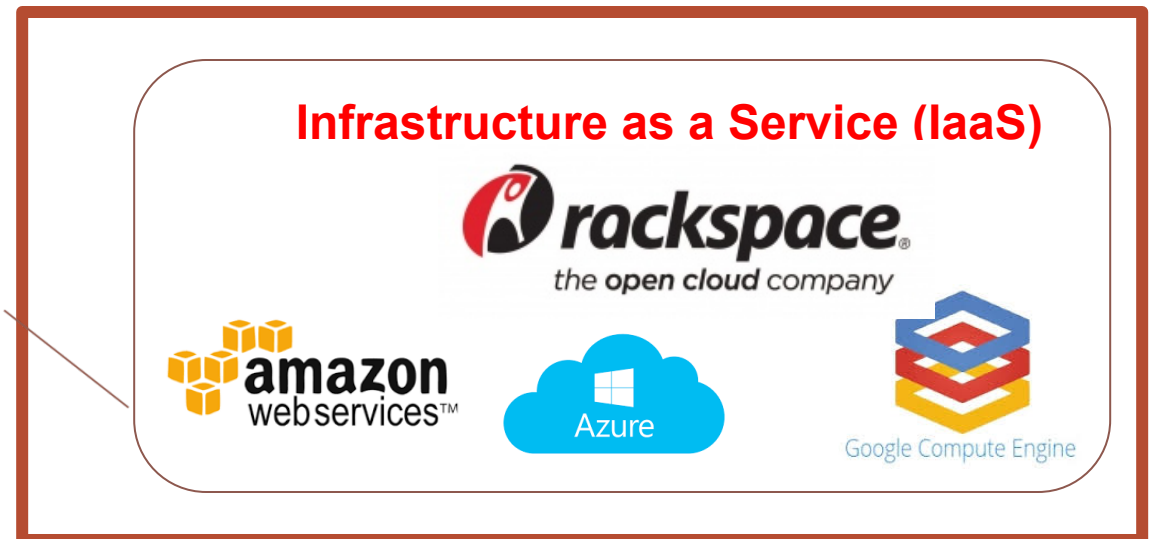
Database-as-a-Service

- ☐ Managed DBMS
- ☐ Relational & NoSQL DBs



IaaS-deployed DBMSs

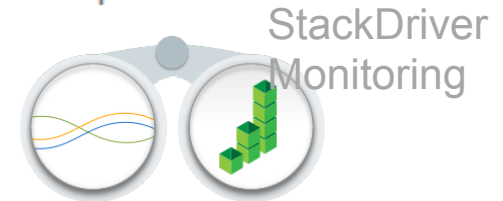
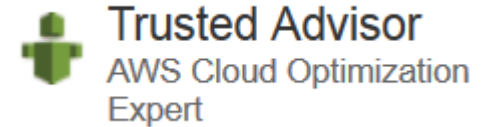
- ☐ Non managed DBMS
- ☐ DIY model



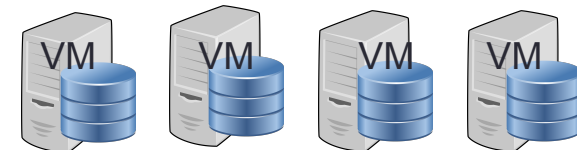
IaaS-deployed Databases

Management Tools

- Monitoring resources, performance, cost
- Event-driven scaling



Data Management Application

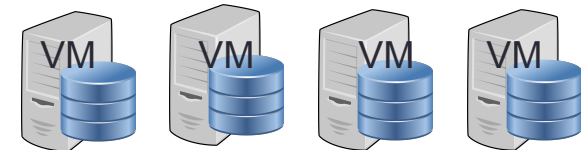


Deployment Challenges



Data Management Application

Custom-built application
management tools



Deployment Challenges



Data Management Application

Cost Management

SLA Management

SLO (objective metric)

- Query-level: response time
- Workload level: average, total, max, percentile

SLA Fees

- Violation penalties



Pay-as-you-go Model



Deployment Challenges



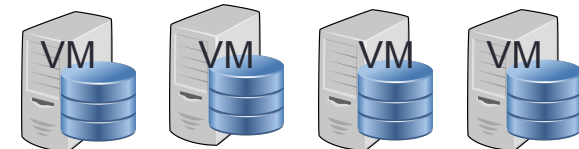
Data Management Application

Cost Management

SLA Management

Resource Provisioning

Workload Scheduling



Beyond monitoring & alerts

- Automatic scale up & down
- Query routing & scheduling
- Cost-driven management
- SLA-awareness

NP-hard problem

State-of-the-art

Placement	Provisioning		Scheduling
PMAX (Liu et al.)	Auto (Rogers et al.)	Dolly (Cecchet et al.)	Shepherd (Chi et al.)
SLATree (Chi et al.)			
Multi-tenant SLOs (Lang et al.)			iCBS (Chi et al.)
Delphi / Pythia (Elmore et al.)	Hypergraph (Çatalyürek et al.)		
SCOPE (Chaiken et al.)	Bazaar (Jalaparti et al.)	many traditional methods ...	

State-of-the-art



Query deadline



Workload deadline



Average latency



Percentile deadline



Piecewise linear

Placement	Provisioning		Scheduling
PMAX (Liu et al.)	Auto (Rogers et al.)	Dolly (Cecchet et al.)	Shepherd (Chi et al.)
SLATree (Chi et al.)			
Multi-tenant SLOs (Lang et al.)			iCBS (Chi et al.)
Delphi / Pythia (Elmore et al.)	Hypergraph (Çatalyürek et al.)		
SCOPE (Chaiken et al.)	Bazaar (Jalaparti et al.)	many traditional methods ...	

Wish List

Requirements

Why ML?

End-to-end cost-aware service
(resource provisioning, workload scheduling)

complex interactions

Application-defined performance goals
(per query deadline, percentile, average latency, max latency)

arbitrary goals

Agnostic to workload semantics

arbitrary workloads

WiSeDB Advisor



Offline Learning

- ❑ batch scheduling
- ❑ performance vs cost exploration

Online Learning

- ❑ online scheduling
- ❑ performance model free

Data Management Application

Cost Management

SLA Management

Resource Provisioning

Workload Scheduling



ORACLE®



Outline

Motivation

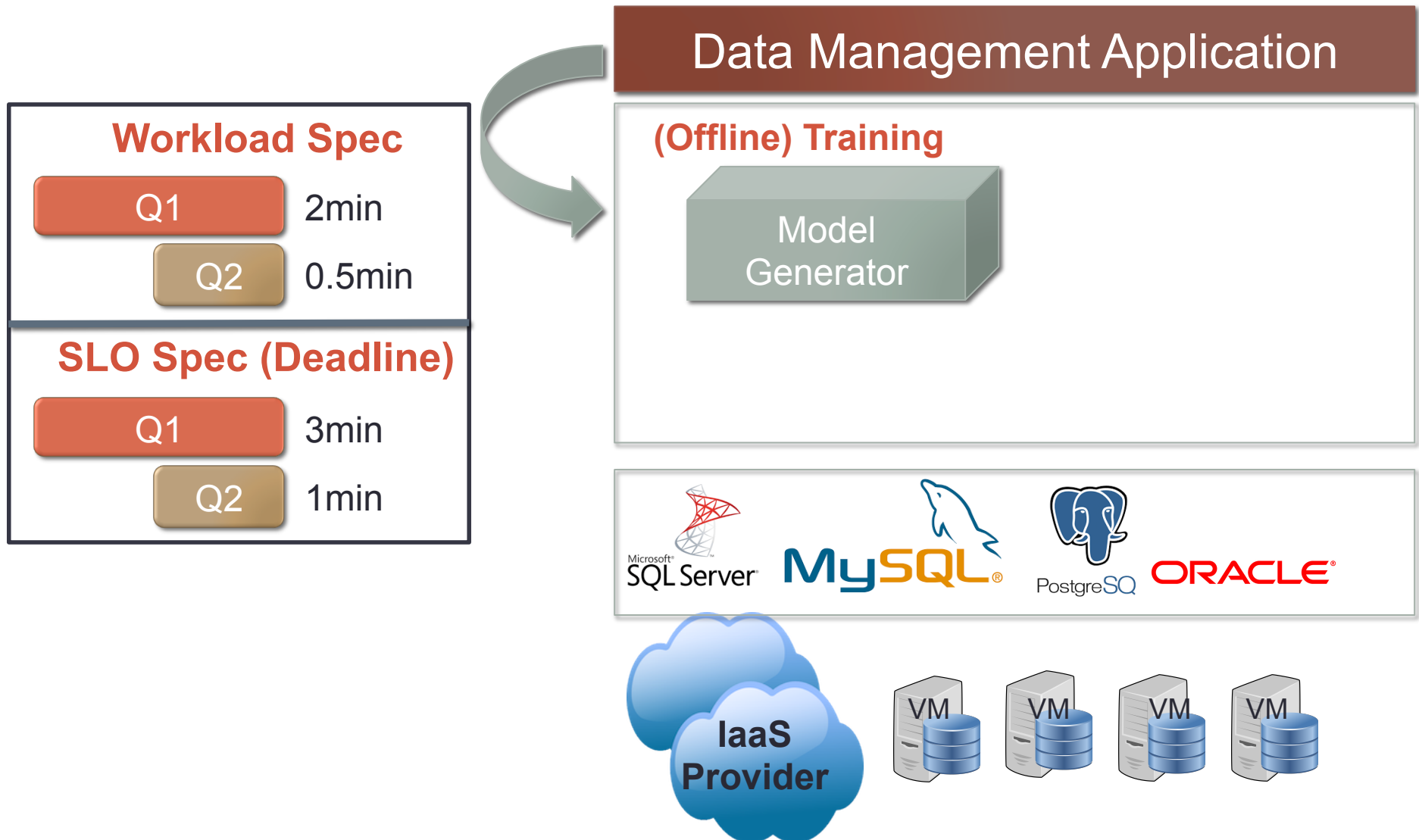
Offline Learning

Online Learning

Conclusions

- System Overview
- Supervised Learning
- Adaptive Learning

Offline Learning



Offline Learning



Original SLO

Q1	3min, \$0.12/Q1
Q2	1min, \$0.2/Q2

Relaxed SLO

Q2	4min, \$0.05/Q1
Q2	2min, \$0.1/Q2

Stricter SLO

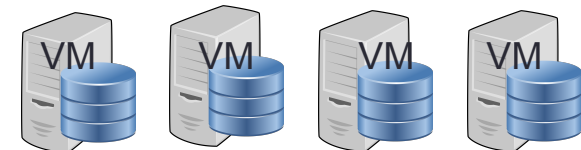
Q1	2.5min, \$0.15/Q1
Q2	0.7min, \$0.13/Q2

Data Management Application

(Offline) Training

Model Generator

Strategy Recommendations



Offline Learning

Relaxed SLO

Q1

4min, \$0.05/Q1

Q2

2min, \$0.1/Q2

Resources to rent

- 2 VMs of Type A
- 3 VMs of Type B

Query scheduling

- VM₁ (Type A) queue
 - Q₁, Q₁, Q₂, Q₂, Q₂,...
- VM₂ (Type B) queue
 - Q₂, Q₂, Q₂, Q₁, Q₁,...

Q1 x 10

Q2 x 200



Data Management Application

(Offline) Training

Model
Generator

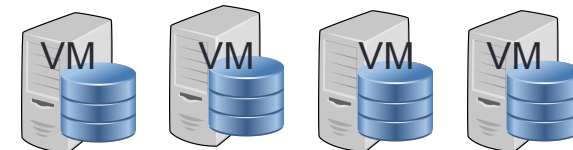
Strategy
Recommendations

(Online) Resource & Workload Management

Strategy
Generator



ORACLE®



Supervised Learning

Model
Generator

identify classes

classes == actions

- dispatch a query to a VM
- provision new VM

create
training data

context of actions

- identify best decisions
- extract cost-related features

generate
classifier

decision tree

- describe (context, action)
- verifiable & interpretable

“To be the best, learn from the best” (D. LaCroix)

Model
Generator

Offline Learning

identify
best decisions

1. Generate small workload
2. Build decision graph
 - ❑ query assignment
 - ❑ VM provisioning
3. Find optimal (minimum cost) solution (path)
4. Extract context of optimal step-by-step decisions

generate
model

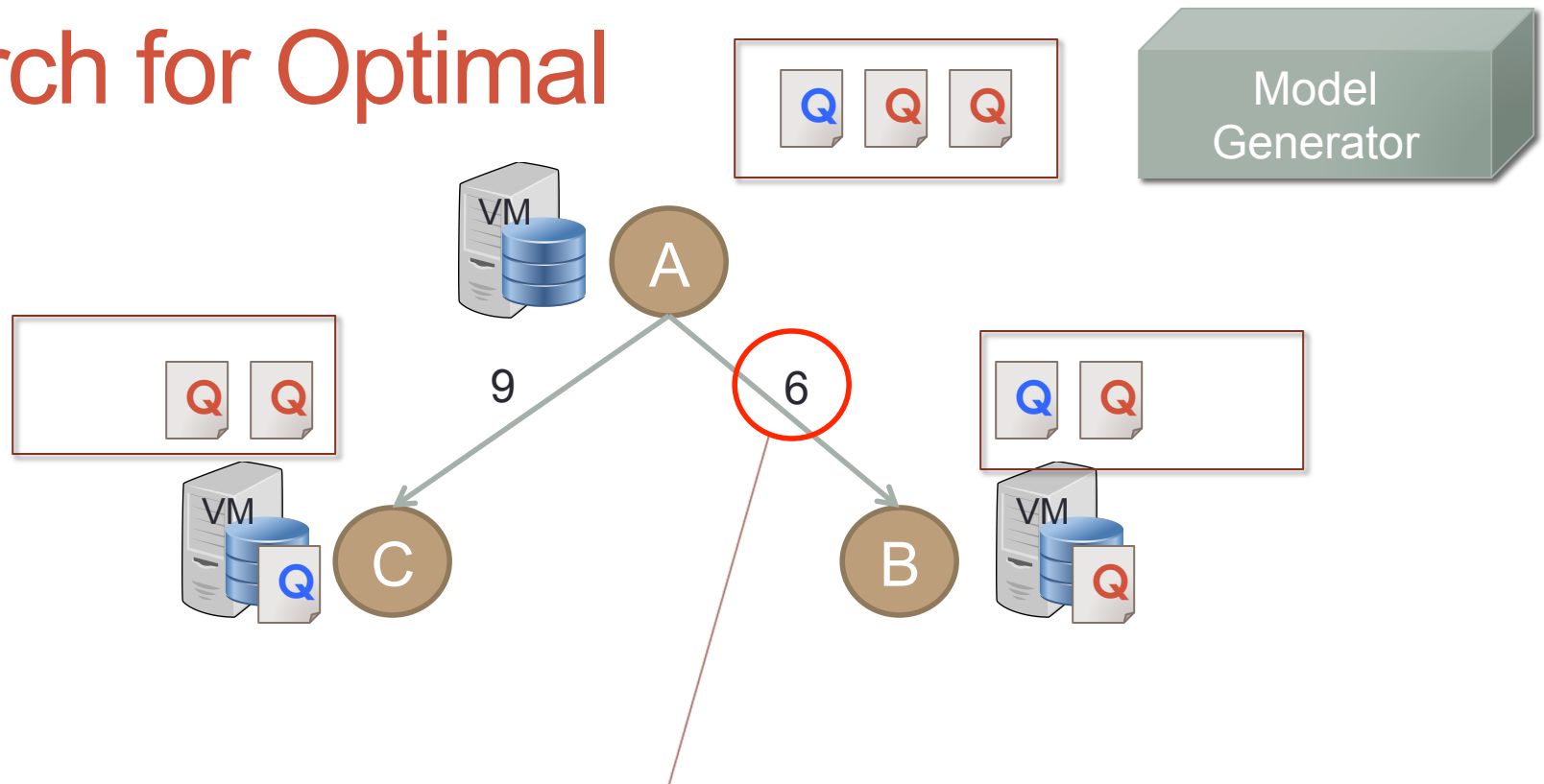
1. Repeat for many sample workloads
2. Build a training set of (feature, action)
3. Train a classifier

Runtime Scheduling

apply
model

- ❑ Use classifier for
 - ❑ batch scheduling
 - ❑ online scheduling
 - ❑ performance vs cost exploration

Search for Optimal

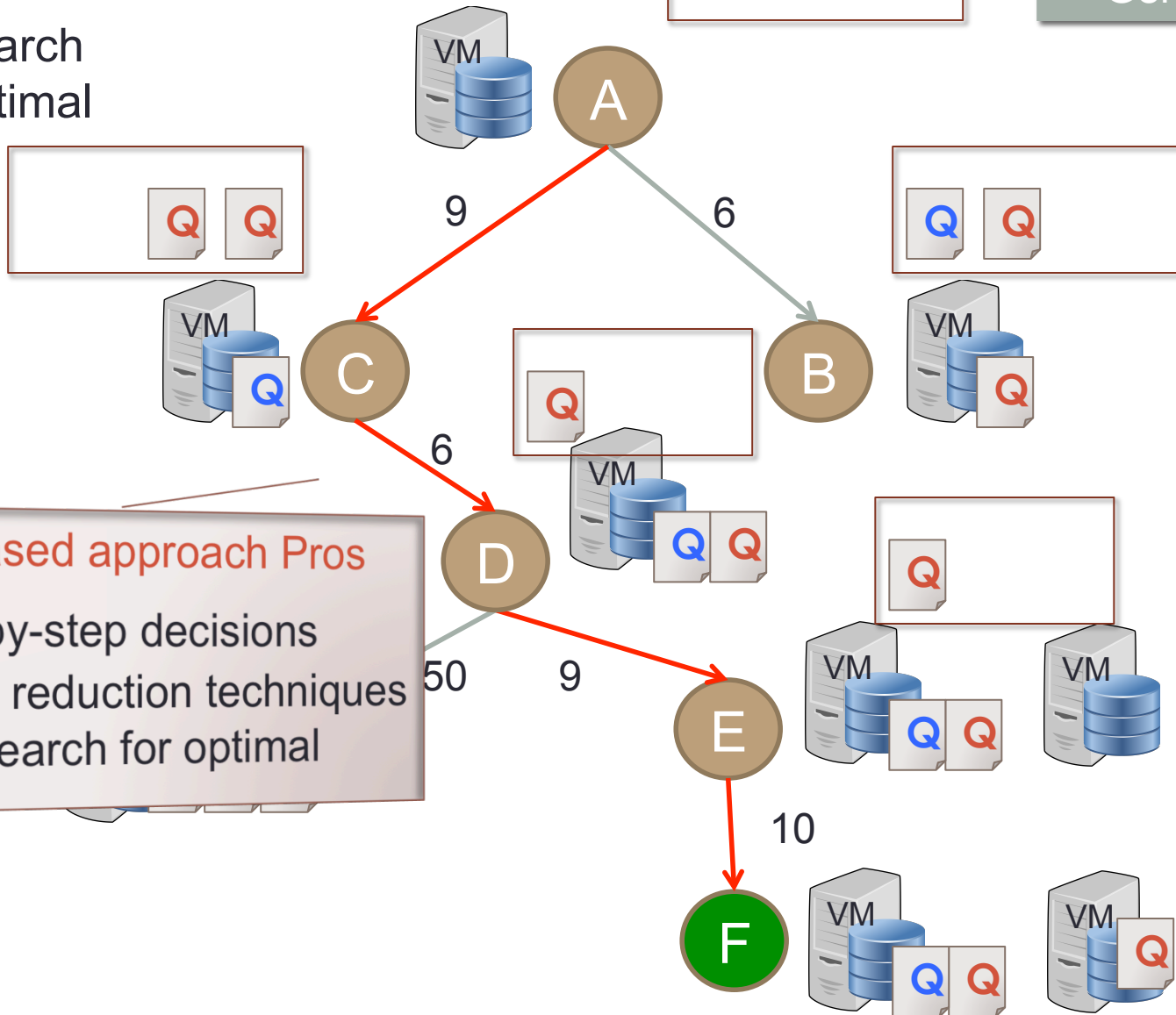
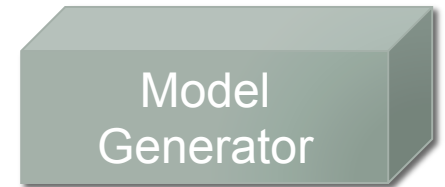


Cost Model

- Resource usage
 - VM start up time + query execution time
- Violation fees
 - Penalty function

Search for Optimal

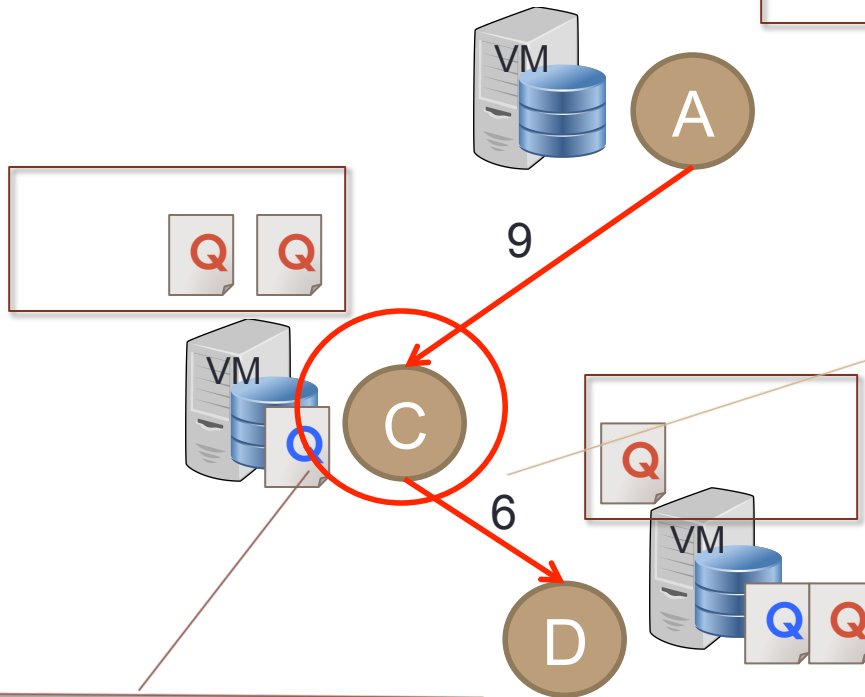
A* search
for optimal




Graph-based approach Pros






- Step-by-step decisions
- Graph reduction techniques
- Fast search for optimal

Feature Extraction



Decision: Assign  to VM

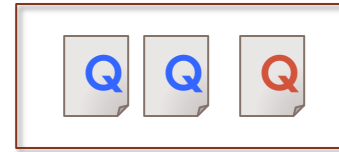
Features:

- unassigned  : true
- unassigned  : false
- cost of assigning  : \$0.2
- wait time on VM: 20sec
- % of  in VM: 0%
- % of  in VM: 0%

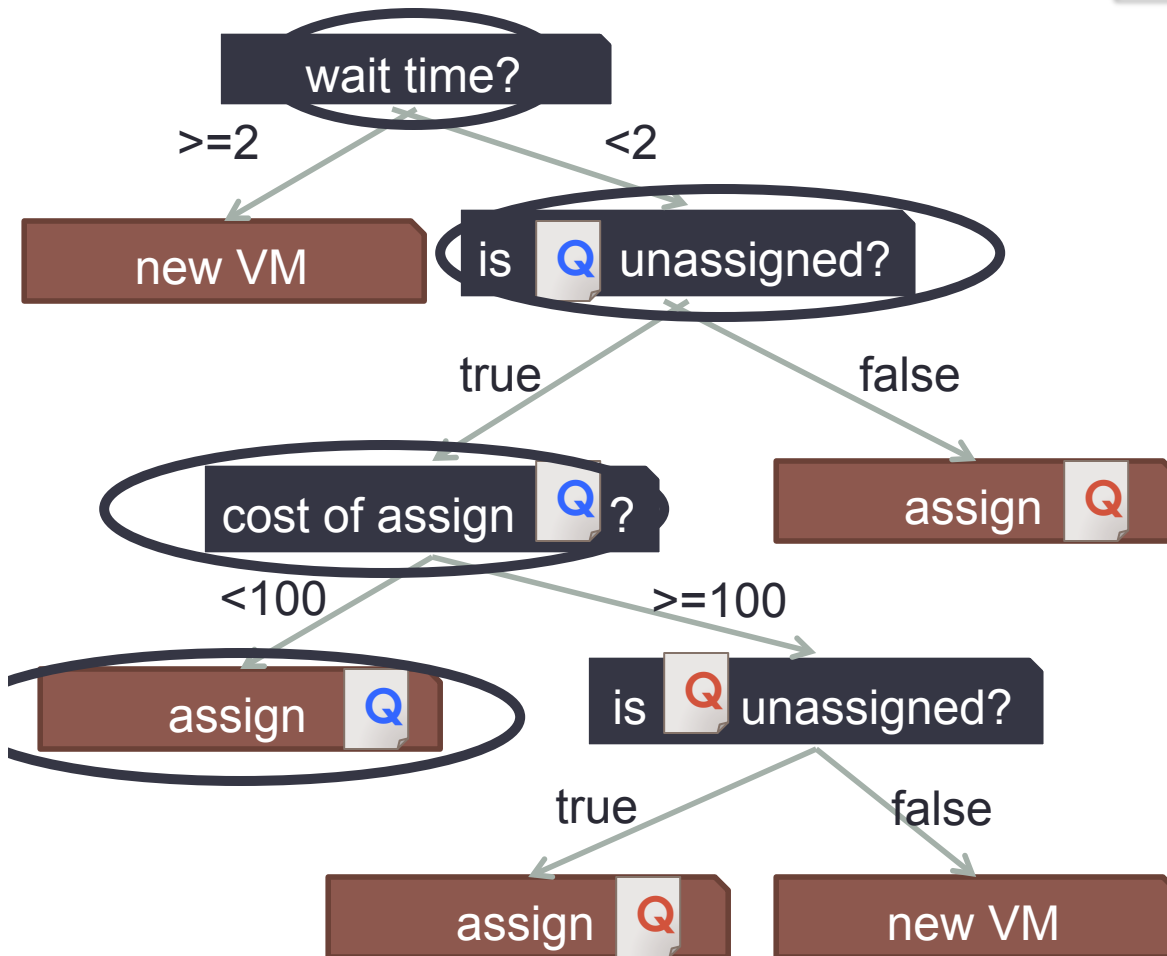
Agnostic to

- Query semantics
- Performance goal (SLO)
- Workload size

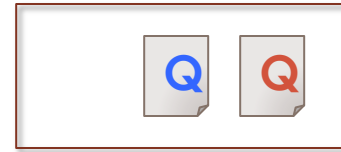
Decision Model



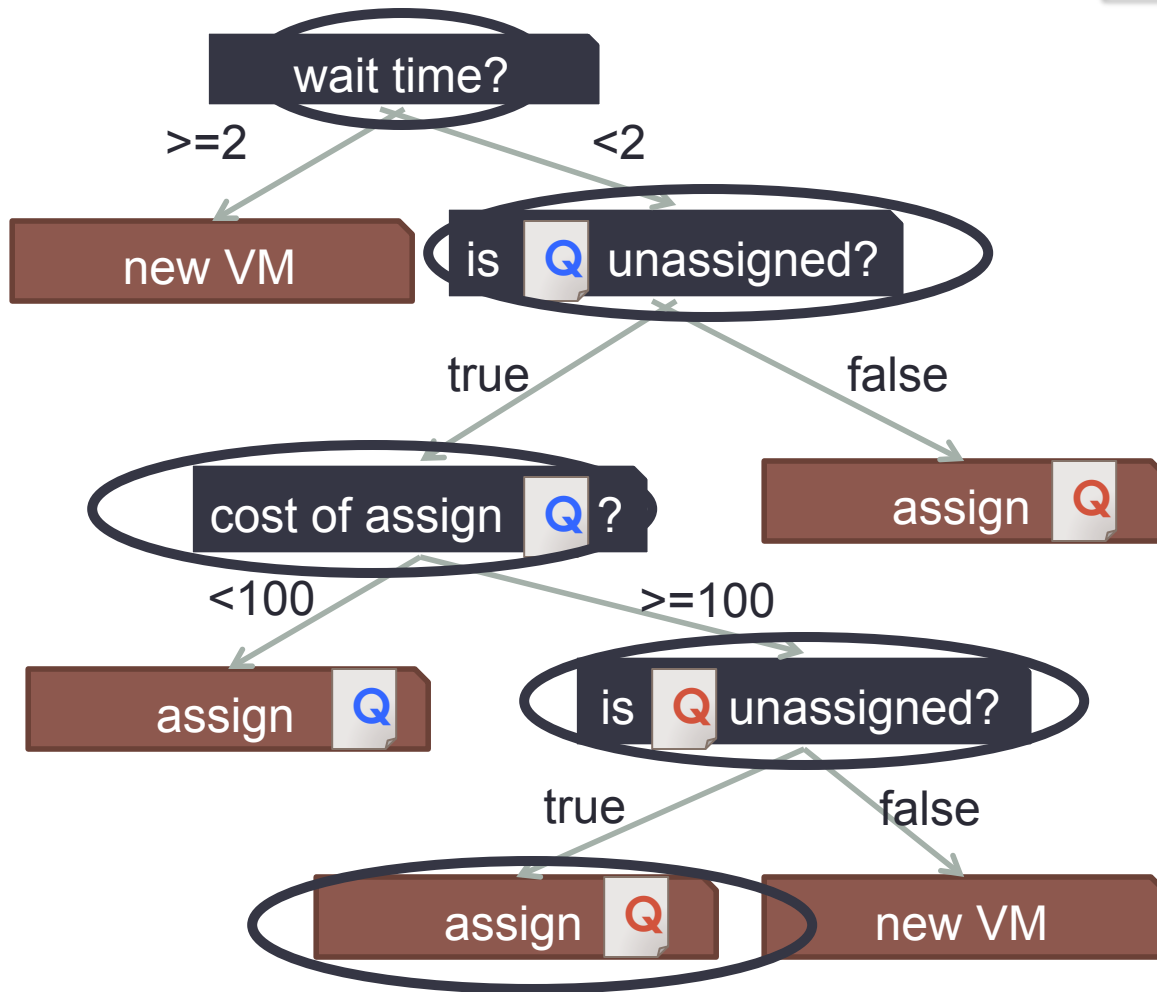
Strategy Generator



Decision Model



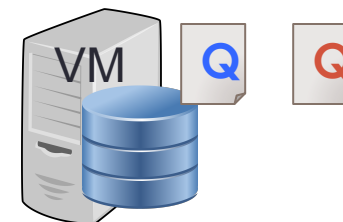
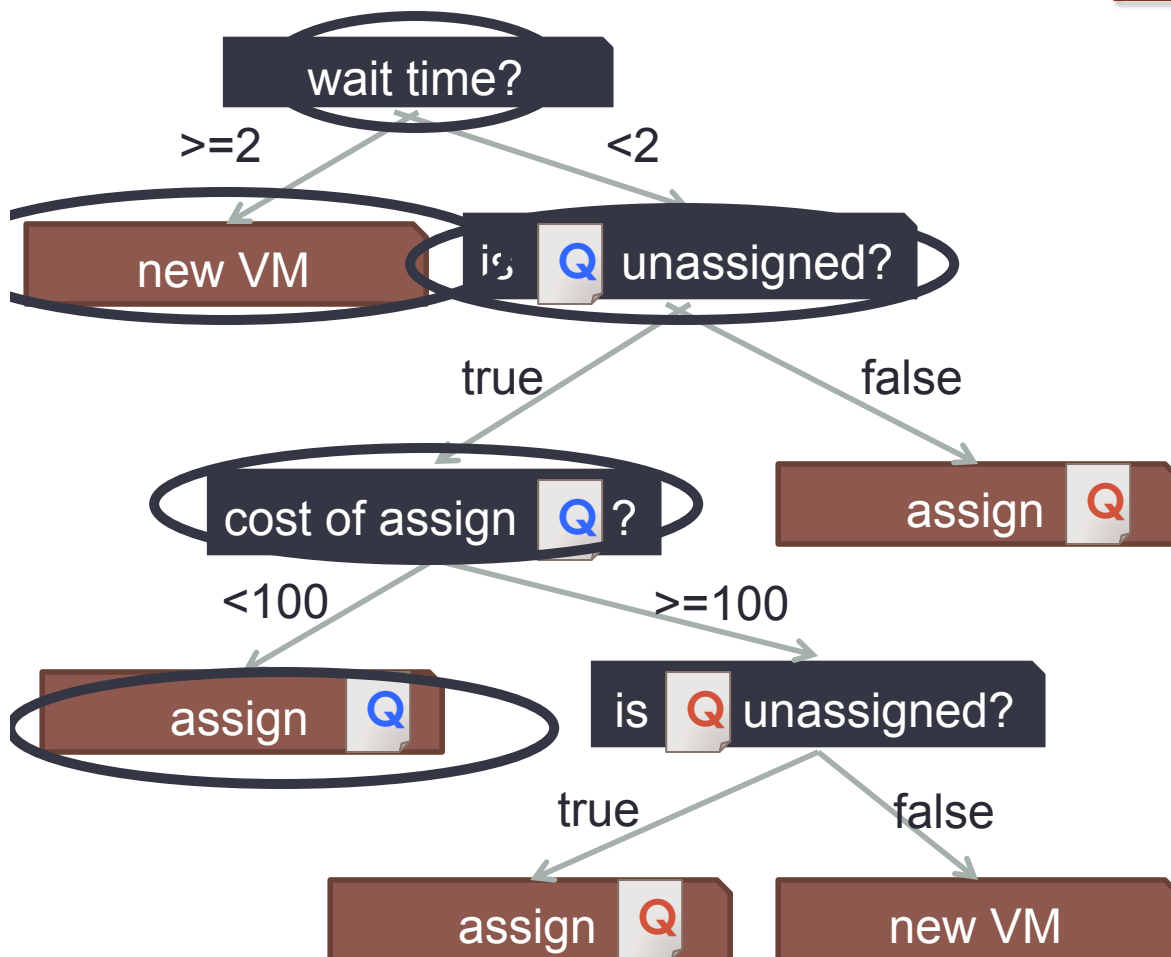
Strategy Generator



Decision Model



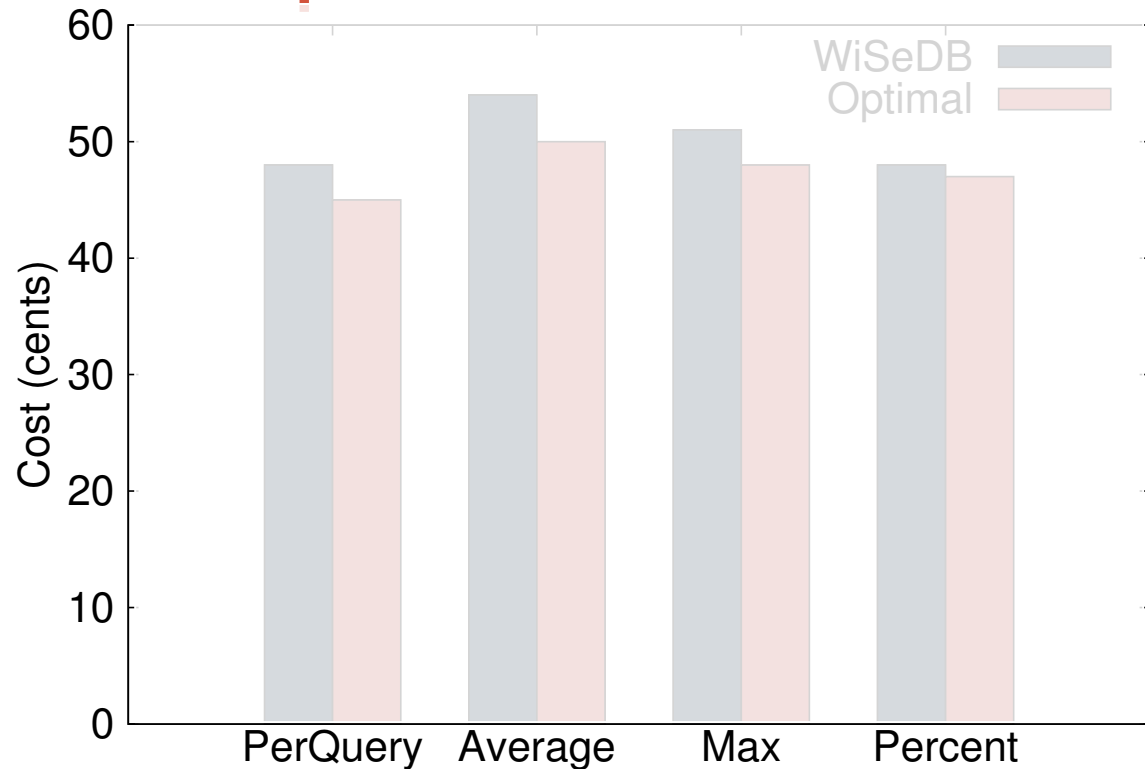
Strategy Generator



Experimental Setup

Training Data

3000 samples
10 TPC-H templates
18 queries/sample

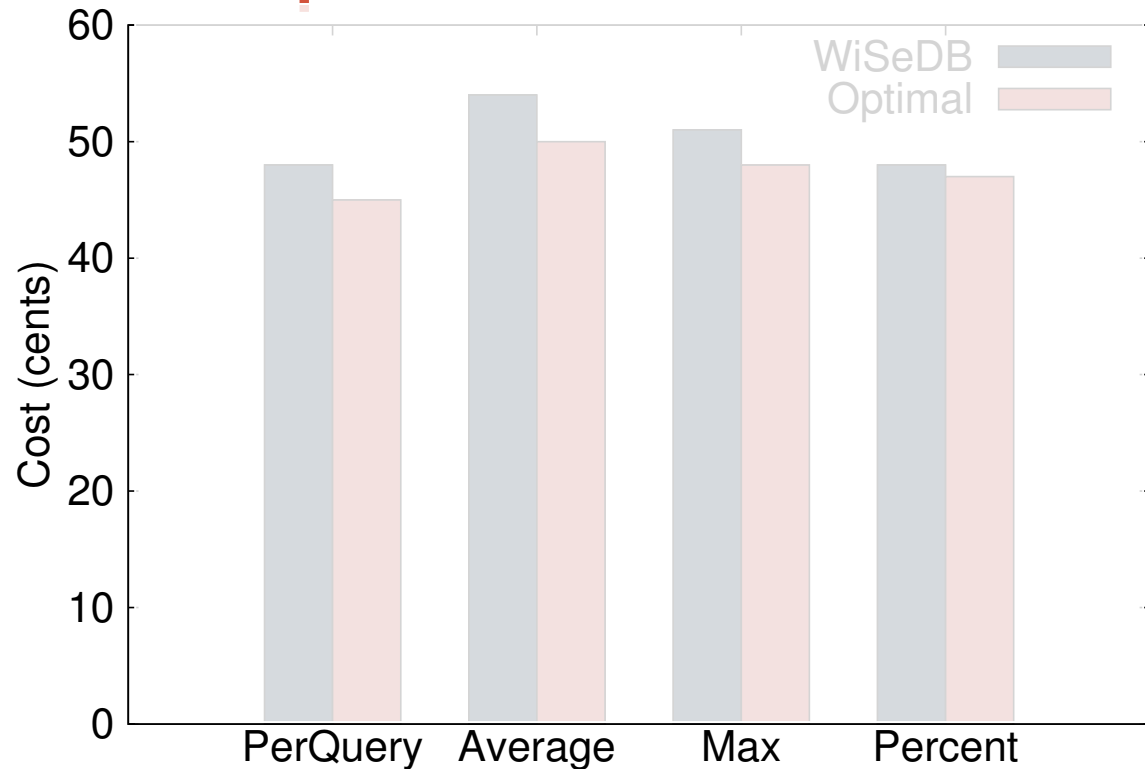


query execution time $\leq x$ secs
(same deadline per template)

Experimental Setup

Training Data

3000 samples
10 TPC-H templates
18 queries/sample

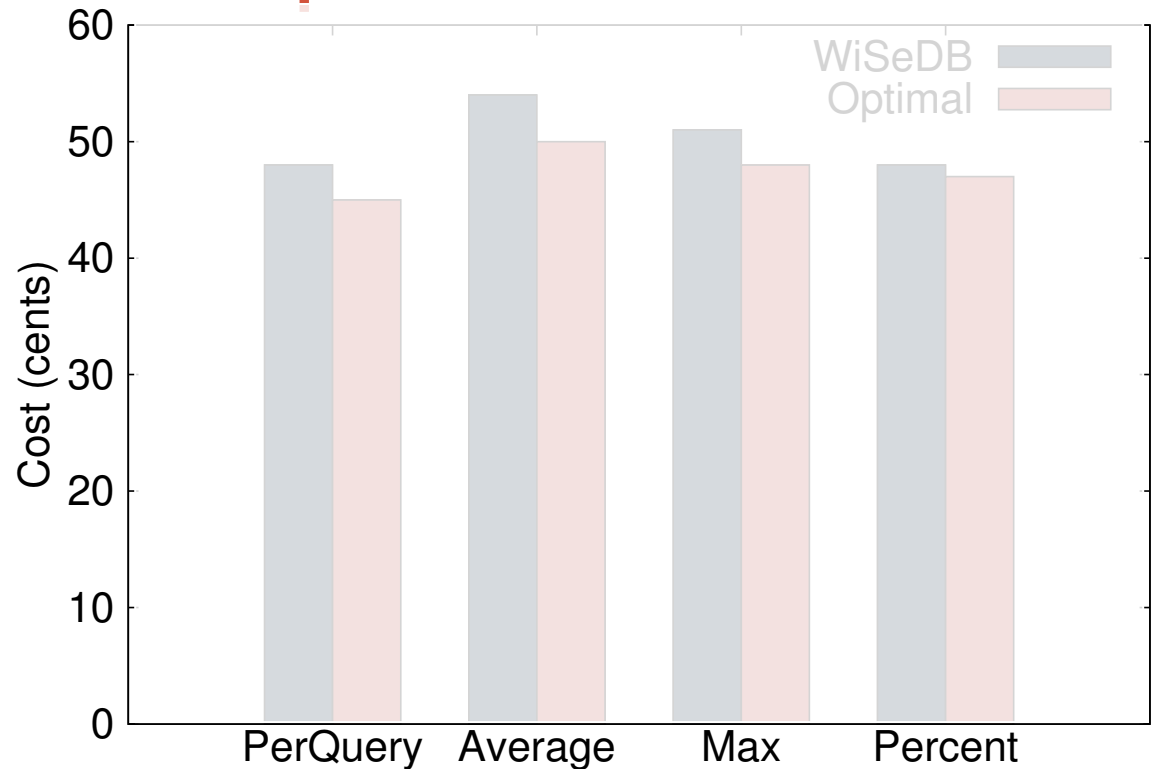


average latency of the
workload $\leq x$ secs

Experimental Setup

Training Data

3000 samples
10 TPC-H templates
18 queries/sample

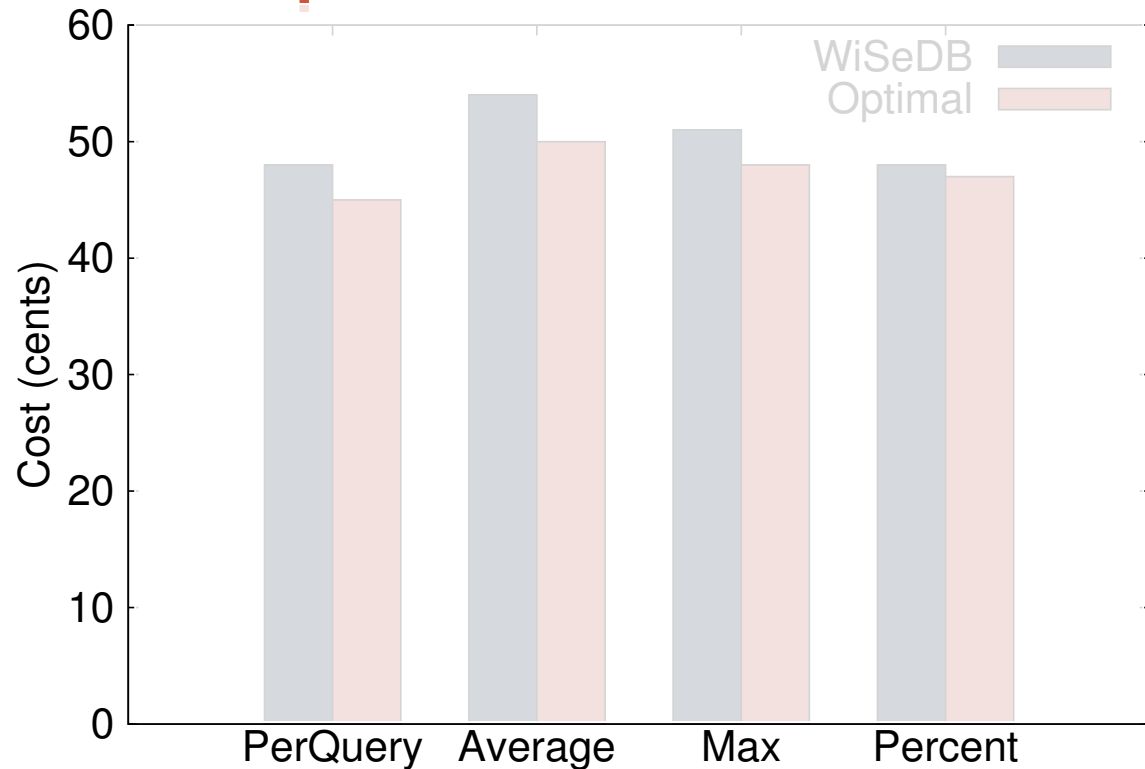


max latency $\leq x$ secs
(longest query in the workload)

Experimental Setup

Training Data

3000 samples
10 TPC-H templates
18 queries/sample



execution time of 90% of queries
in the workload $\leq x$ secs

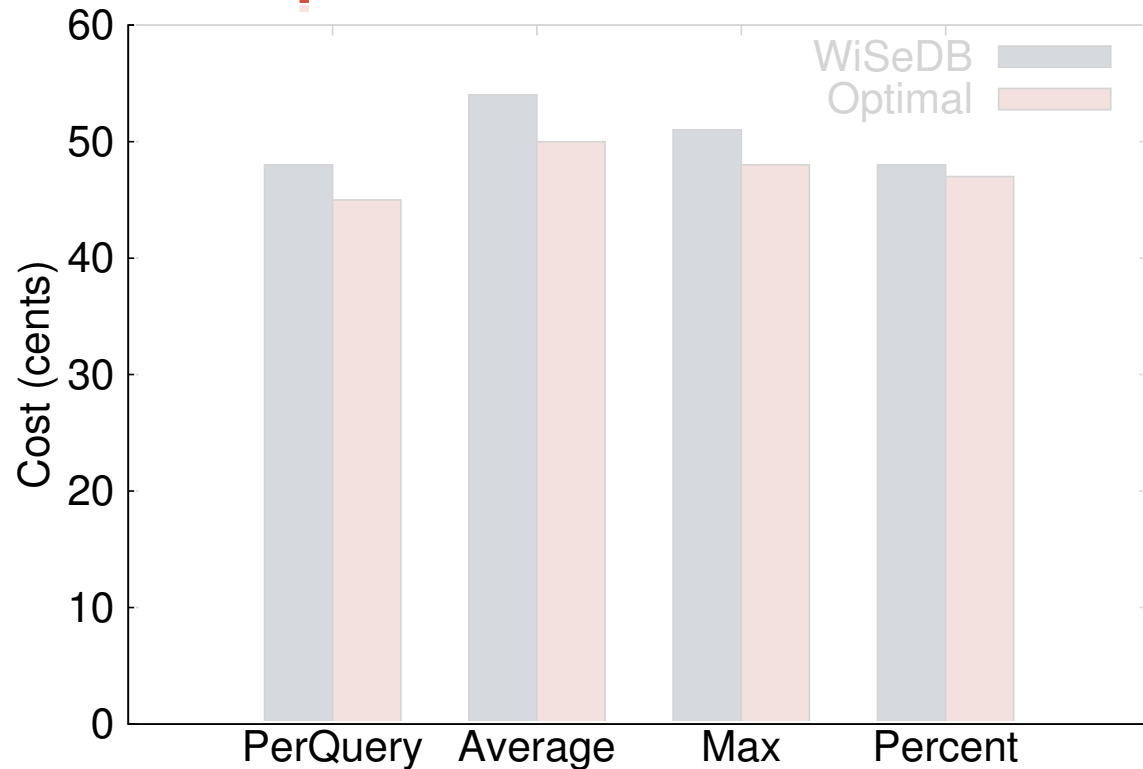
Experimental Setup

Training Data

3000 samples
10 TPC-H templates
18 queries/sample

Testing Data

10 TPC-H templates
varied queries/workload



Experimental Setup

Training Data

3000 samples
10 TPC-H templates
18 queries/sample

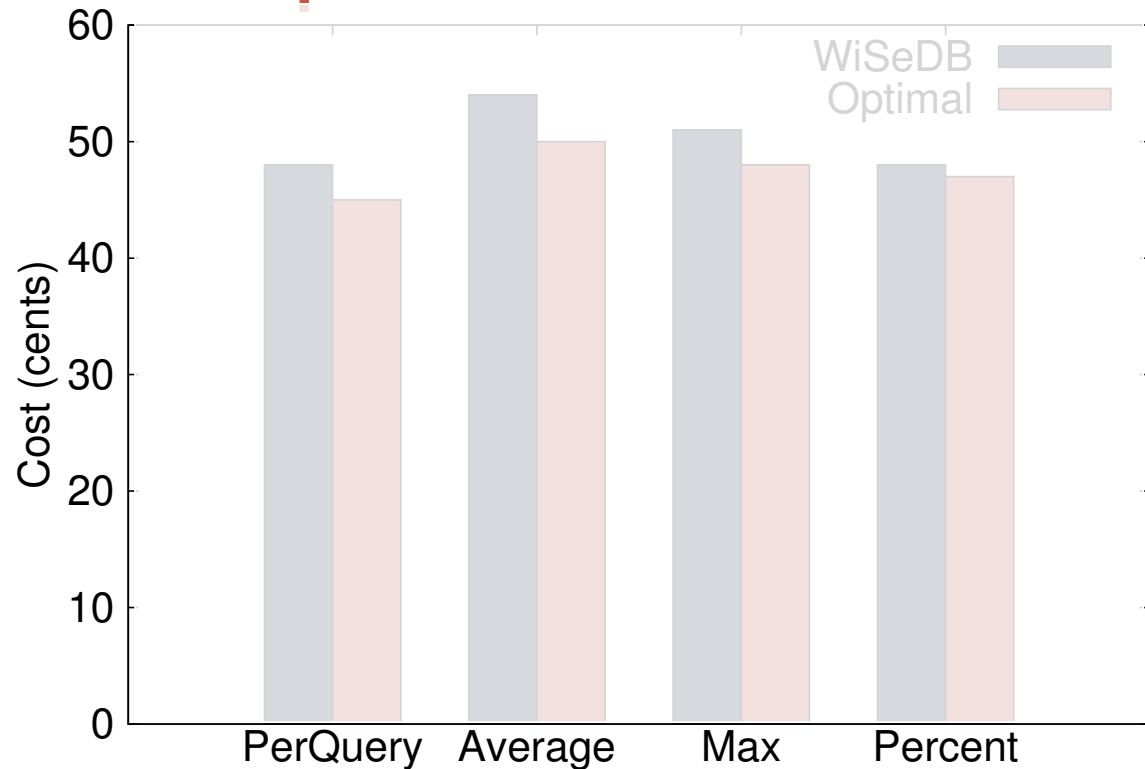
Testing Data

10 TPC-H templates
varied queries/workload

cost: resource utilization + penalties

AWS Cloud

fees penalty \$0.01/sec of violation



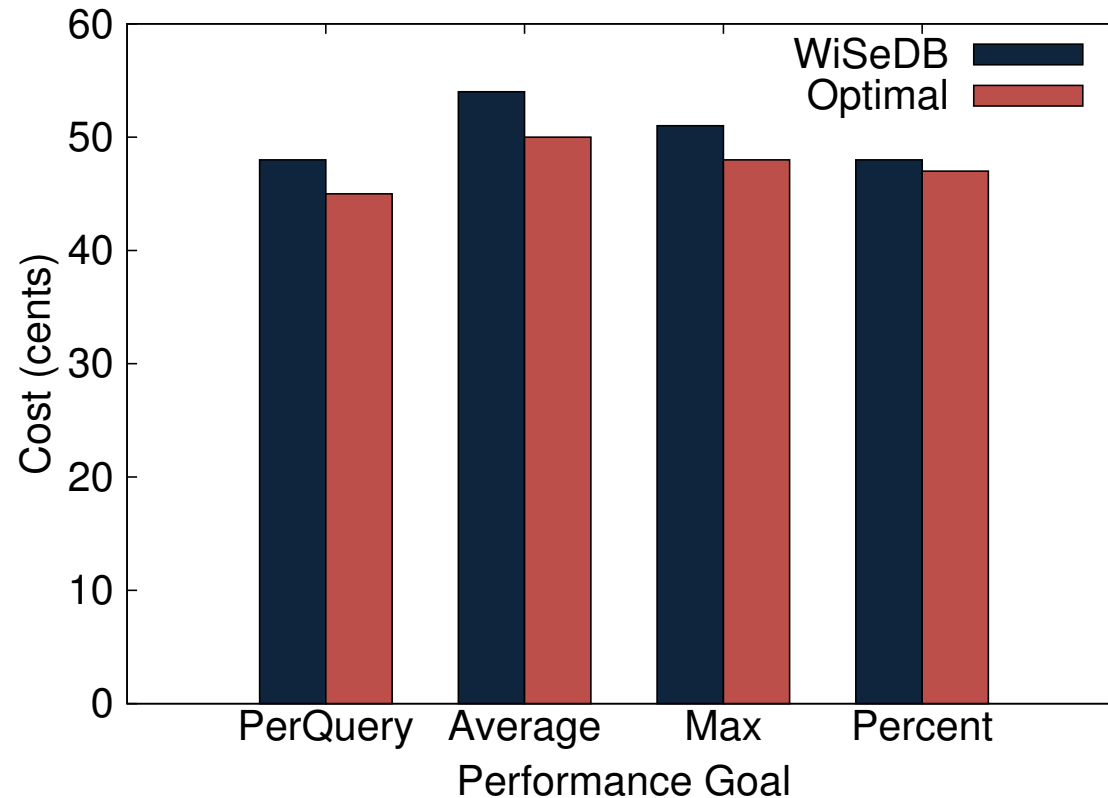
Effectiveness (small workloads)

Training Data

3000 samples
10 TPC-H templates
18 queries/sample

Testing Data

10 TPC-H templates
30 queries/workload
Optimal: Brute force



WiSeDB models are within 8% of the minimum cost solution

Effectiveness (large workloads)

Training Data

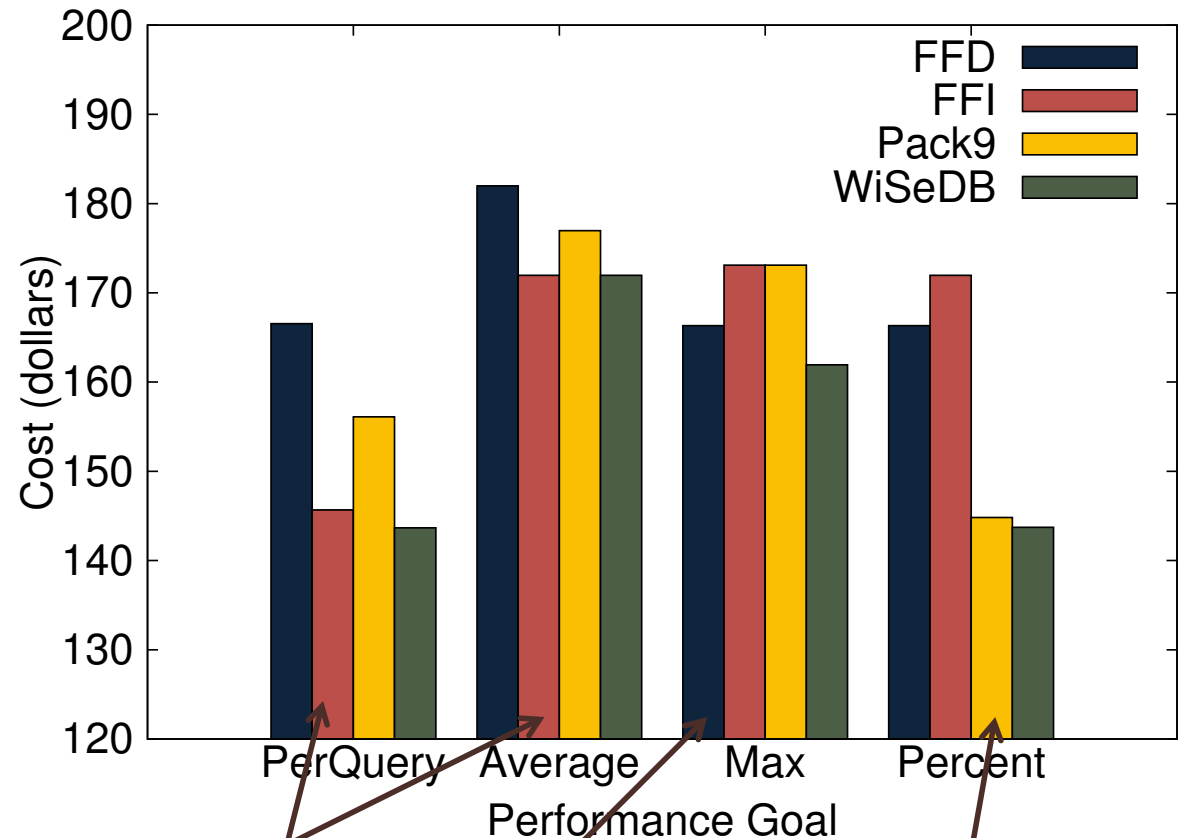
3000 samples
10 TPC-H templates
18 queries/sample

Testing Data

10 TPC-H templates
5000 queries/workload

One heuristic cannot fit all

WiSeDB learns the right heuristic



Best: shortest query first

Best: longest query first

Best: top-90% shortest then 10% longest queries

Offline Learning



Original SLO

Q1	3min, \$0.12/Q1
Q2	1min, \$0.2/Q2

Relaxed SLO

Q1	4min, \$0.05/Q1
Q2	2min, \$0.1/Q2

Stricter SLO

Q1	2.5min, \$0.15/Q1
Q2	0.7min, \$0.13/Q2

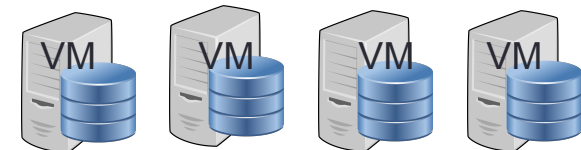
Data Management Application

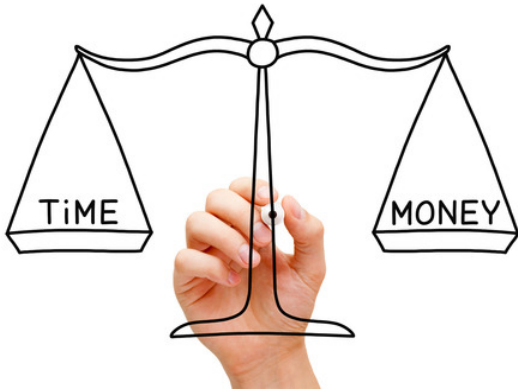
(Offline) Training

Model Generator

Strategy Recommendations

(Online) Performance Management





exploration of
performance vs cost
trade offs

Strategy
Recommendations

new SLO

new scheduling graph

new optimal decisions (path)

new model

expensive
(brute force/sample)

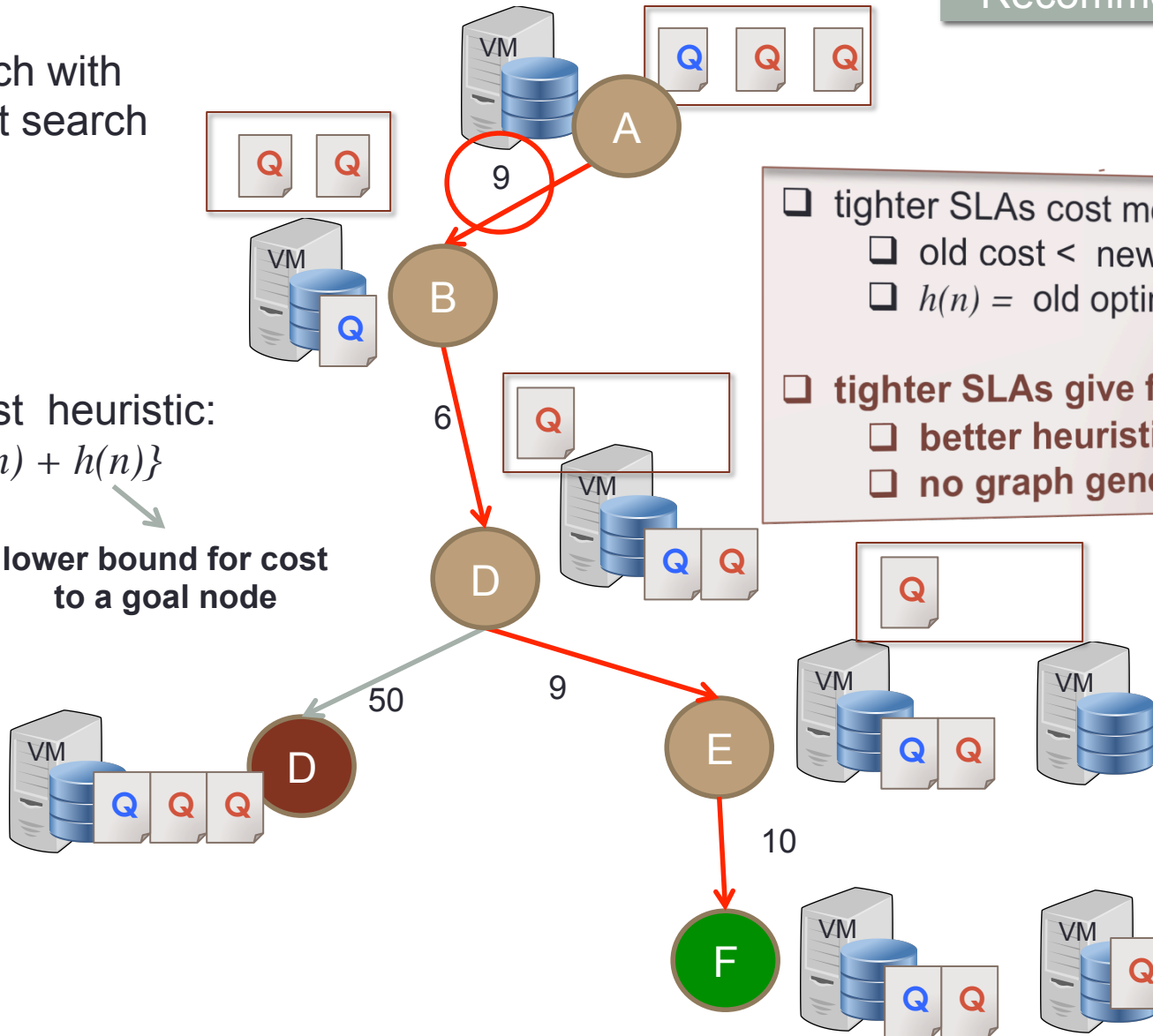
change only the SLO & reuse the original graph

Adaptive Modeling

Strategy Recommendations

Fast search with A* best-first search

explore-first heuristic:
 $\min \{g(n) + h(n)\}$
cost so far lower bound for cost to a goal node



- tighter SLAs cost more
 - old cost < new cost
 - $h(n) = \text{old optimal cost}$
- tighter SLAs give faster search
 - better heuristic
 - no graph generation

Adaptive Training

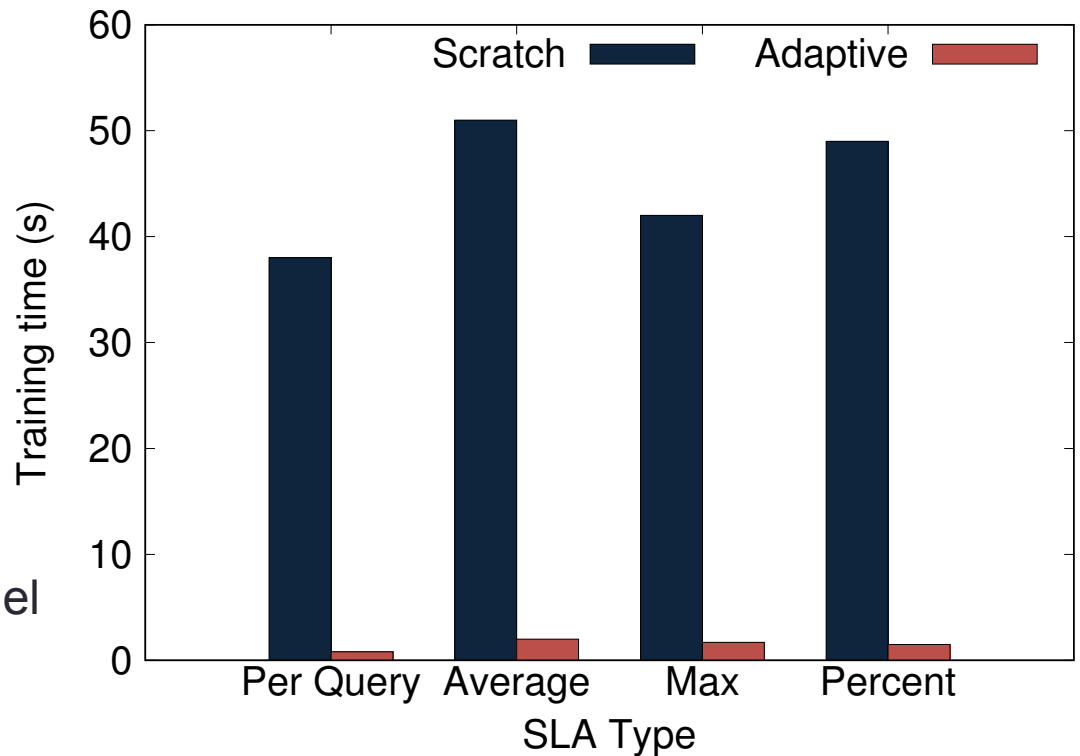
Training Data

3000 samples
10 TPC-H templates
18 queries/sample

15% stricter SLA

Scratch: training a new model

Adaptive: adapting the original model



Adaptive training time is 96-94% less than original training time

Performance vs Cost Exploration

Strategy
Recommendations

Original SLO

Q1 3min, \$0.12/Q1

Q2 1min, \$0.2/Q2

Relaxed SLO

Q1 4min, \$0.05/Q1

Q2 2min, \$0.1/Q2

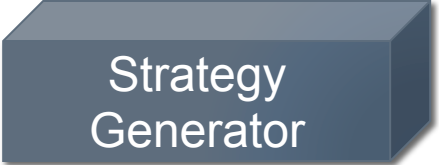
Stricter SLO

Q1 2.5min, \$0.15/Q1

Q2 0.7min, \$0.13/Q2

- ❑ WiSeDB generates models for 10s of alternative SLOs within secs
 - ❑ Keeps k-top significant ones
 - ❑ Earth Mover's Distance
 - ❑ No query execution is required
- ❑ Model estimates cost/template & expected performance
 - ❑ Assumes a given cost model
- ❑ User picks desired model

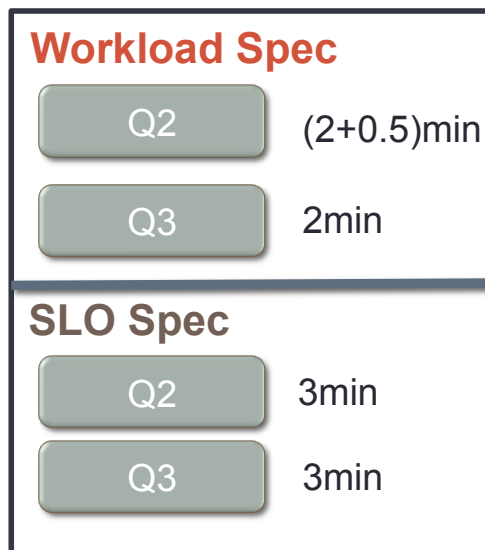
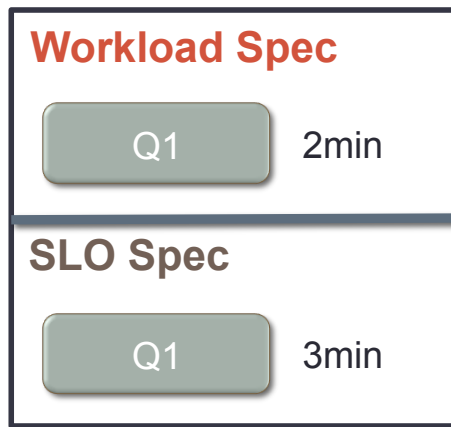
Online Scheduling

A dark blue 3D rectangular box with the text "Strategy Generator" in white, positioned in the top right corner of the slide.

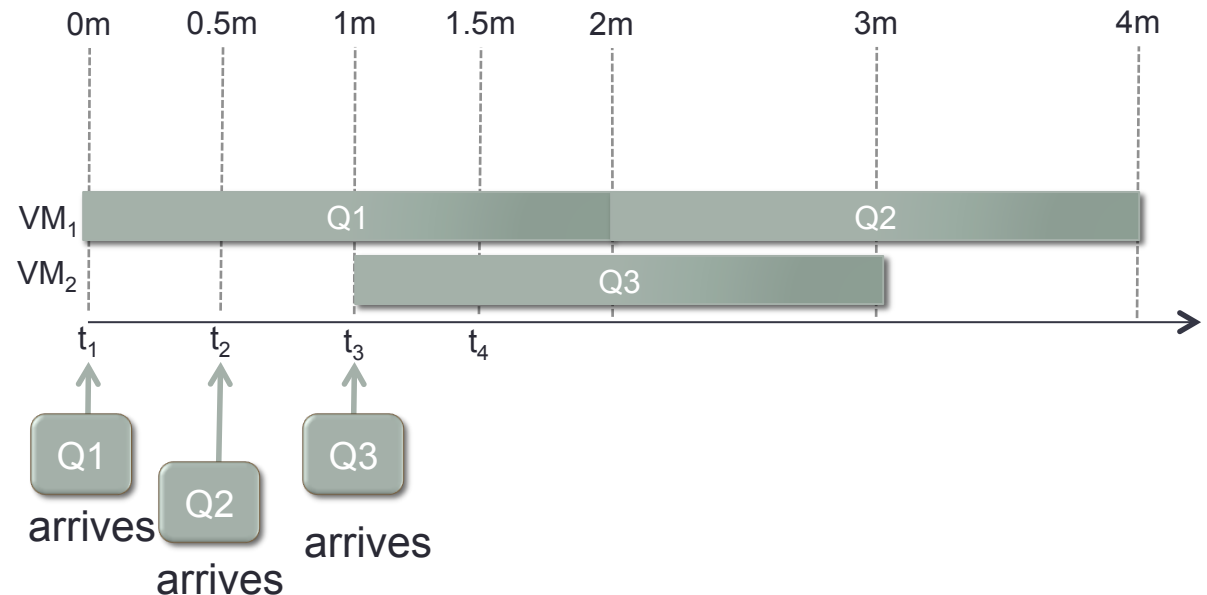
Strategy
Generator

- Scheduling & provisioning for one query at a time
- Batch-based models not effective for online tasks
 - Do not account for query arrival rate/wait times
- WiSeDB approach
 - Generate a new model upon arrival of new query
 - Adapt previous model to reduce training overhead
 - Reuse past models, when feasible

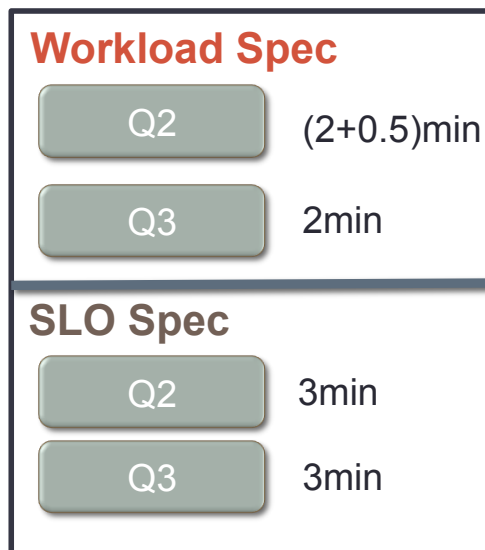
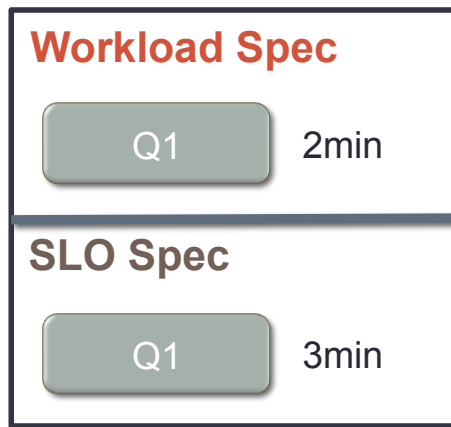
Online Scheduling



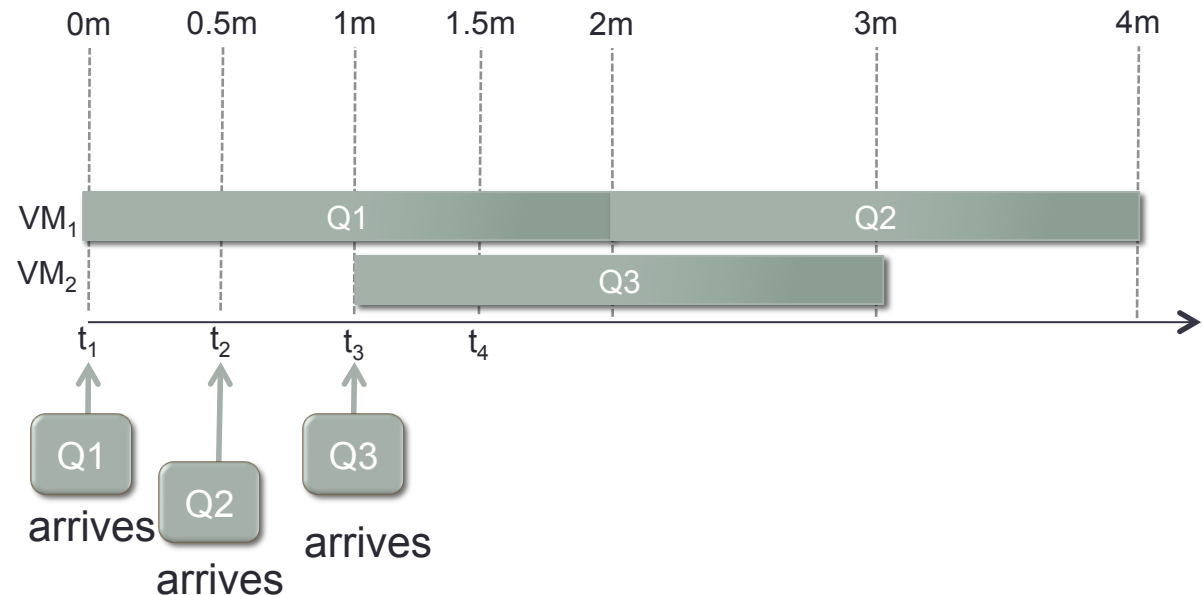
- ❑ training batch: new query + queued queries
- ❑ add wait time in expected latency
- ❑ slow for for high arrival rates



Online Scheduling



- ❑ Model Reuse: reuse model with similar expected latencies/template
- ❑ Linear Shifting: treat as a tightened SLA

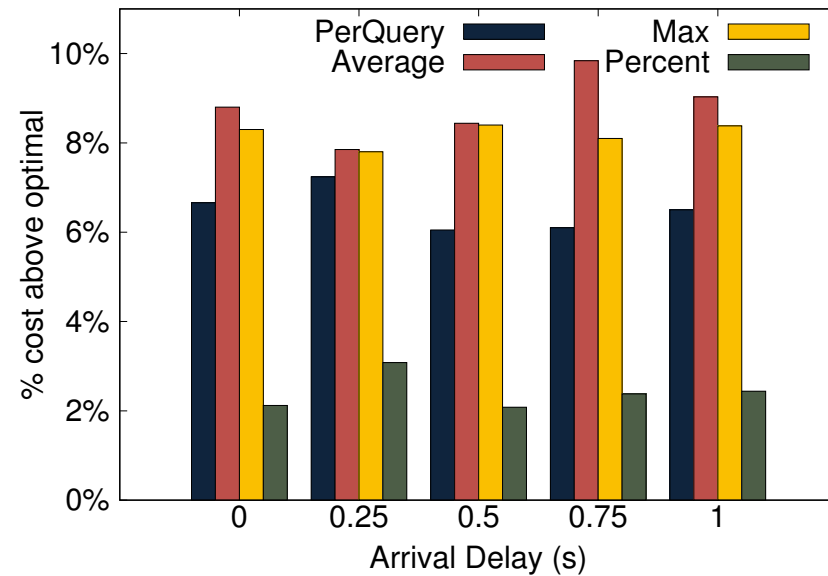
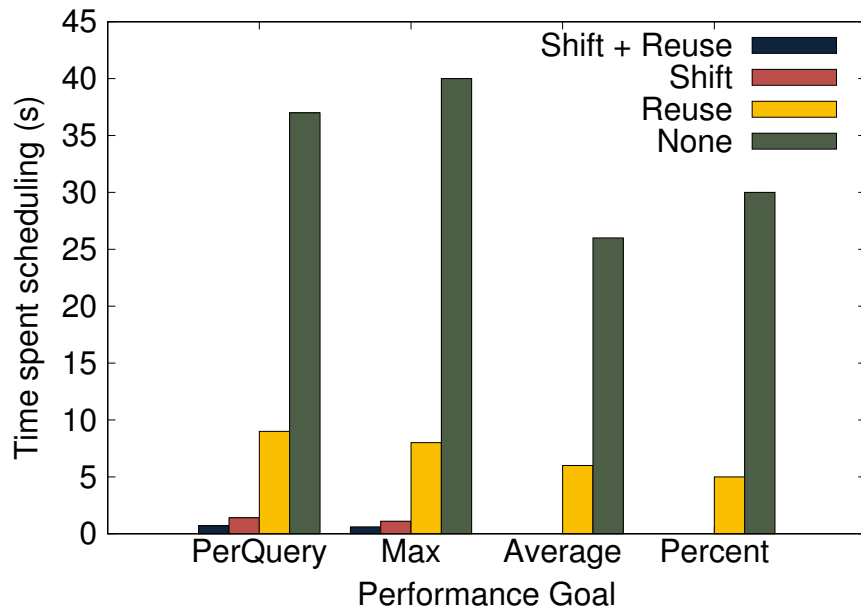


Effectiveness (online scheduling)

Testing Data

30 queries/workload
10% from optimal

Query wait time < 1 sec



WiSeDB can leverage existing models to offer effective scheduling in a online manner

Offline Learning



Advantages

- ❑ Abstracts away complex decisions
- ❑ Generates custom heuristics per application
- ❑ Explores Performance vs Cost trade-offs

Data Management Application

(Offline) Training

Model
Generator

Strategy
Recommendations

(Online) Resource & Workload Management

Strategy
Generator



ORACLE®

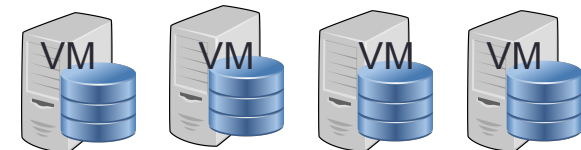
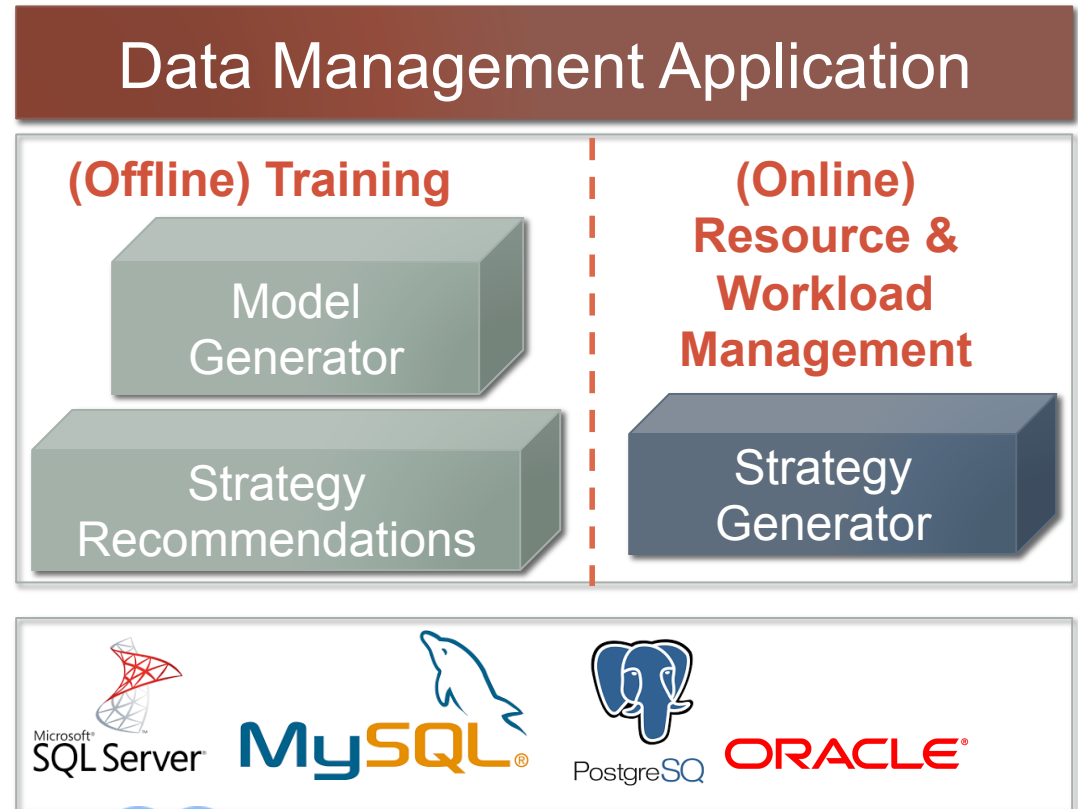


Offline Learning



Limitations

- Static models
- Batch scheduling
- Known cost model



Outline

Motivation

Offline Learning

Online Learning

Conclusions

- Explicit vs Implicit Modeling
- Reinforcement Learning

(Explicit) Performance Prediction

- ❑ DBMS-related challenges
 - ❑ isolated vs. concurrent query execution
 - ❑ low accuracy for new query types (“templates”)
 - ❑ extensive off-line training
 - ❑ **state-of-the-art: 15-20% prediction error**
- ❑ Cloud-related challenges
 - ❑ “noisy neighbors”
 - ❑ numerous resource configurations
 - ❑ predictions errors accumulation

WiSeDB: Implicit Performance Modeling

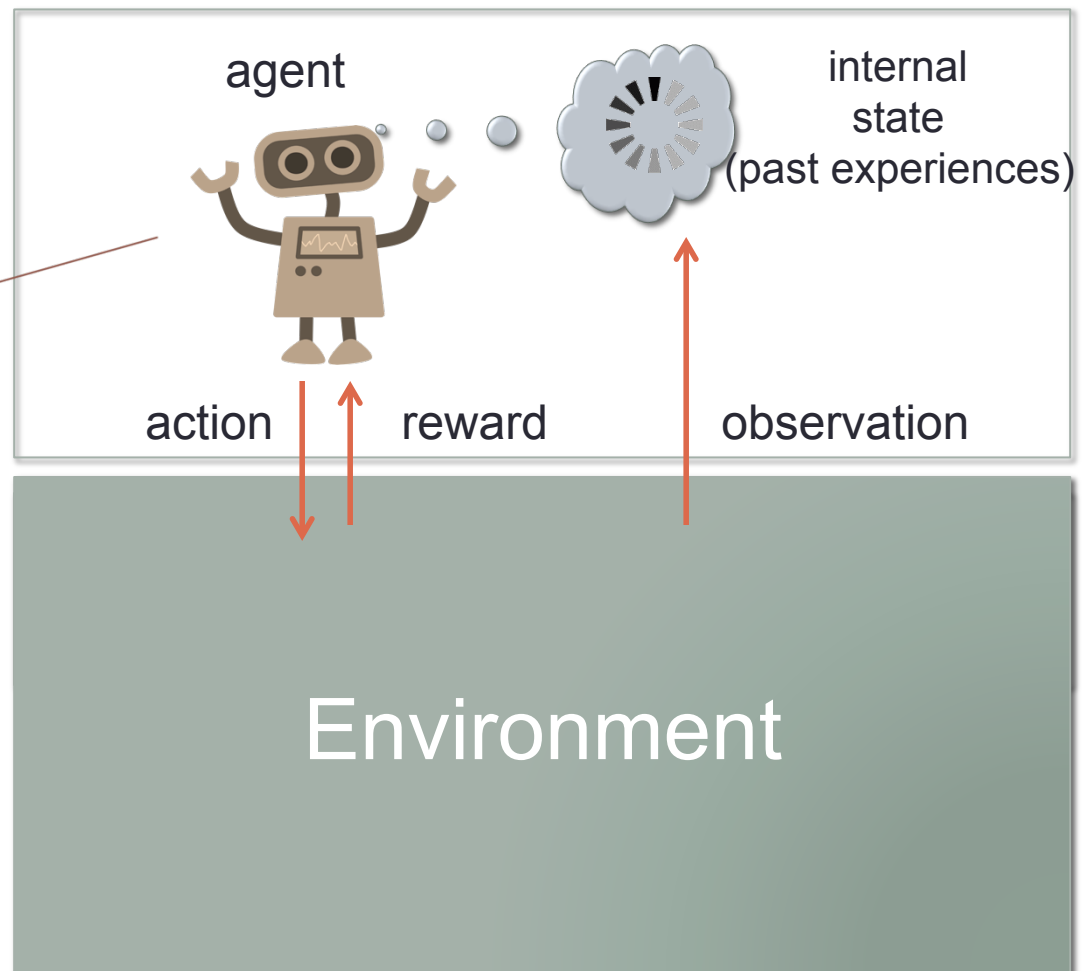
- ❑ **Explicit performance models are NOT necessary for:**
 - ❑ monetary cost management
 - ❑ resource & workload management
 - ❑ offer performance SLA and keep penalties low

Wish List #2

- ❑ Implicitly model query latency
 - ❑ predict *monetary cost (& violation penalties)*
- ❑ Online training for dynamic environments
 - ❑ Automatic scaling & workload distribution

Reinforcement Learning

- Continuous learning
- Explicit reward modeling
- Action selection
 - maximize reward



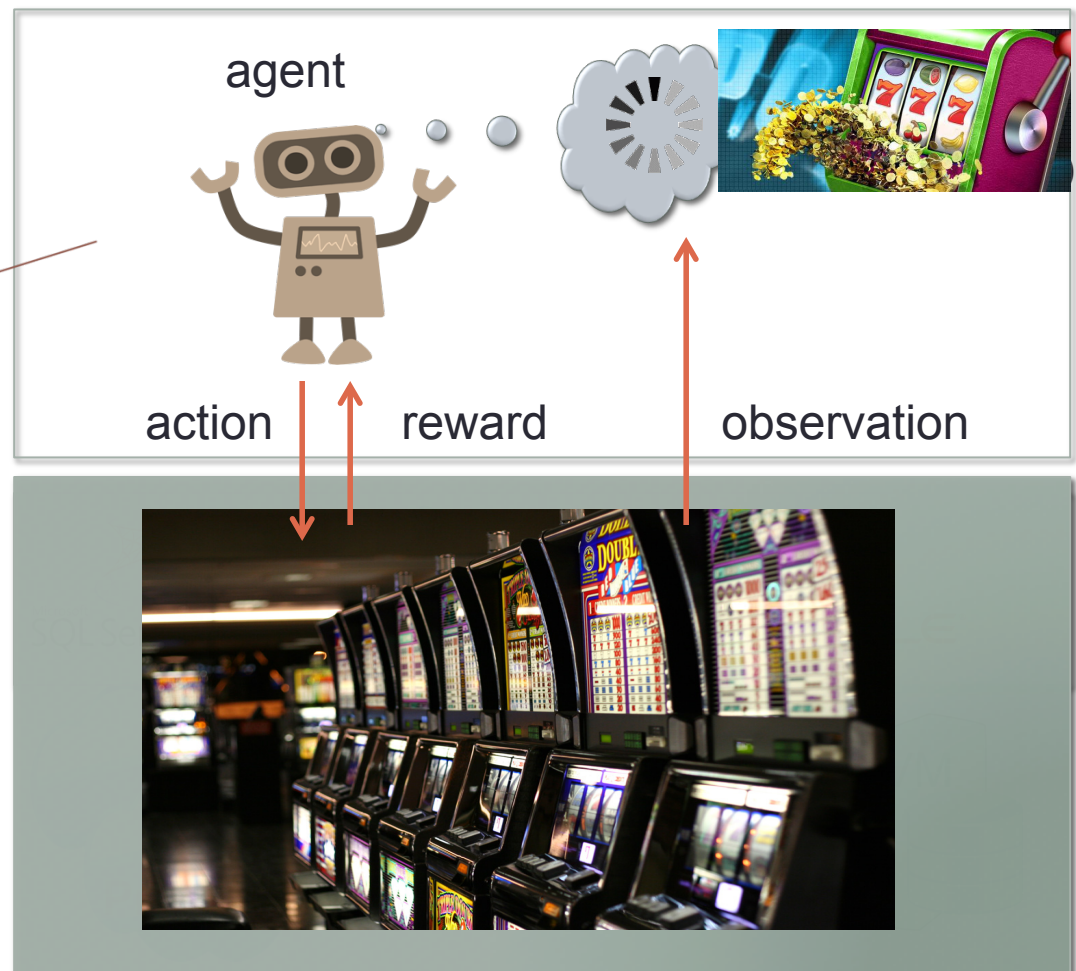
CMABs

(Contextual Multi-Armed Bandits)

Contextual Multi-Armed Bandit Problem

Armed Bandit = Slot Machine

*Which slot machine to play (**action**) so that you walk out with the most \$\$\$ (**reward**)?*



CMABs in WiSeDB

(Contextual Multi-Armed Bandits)

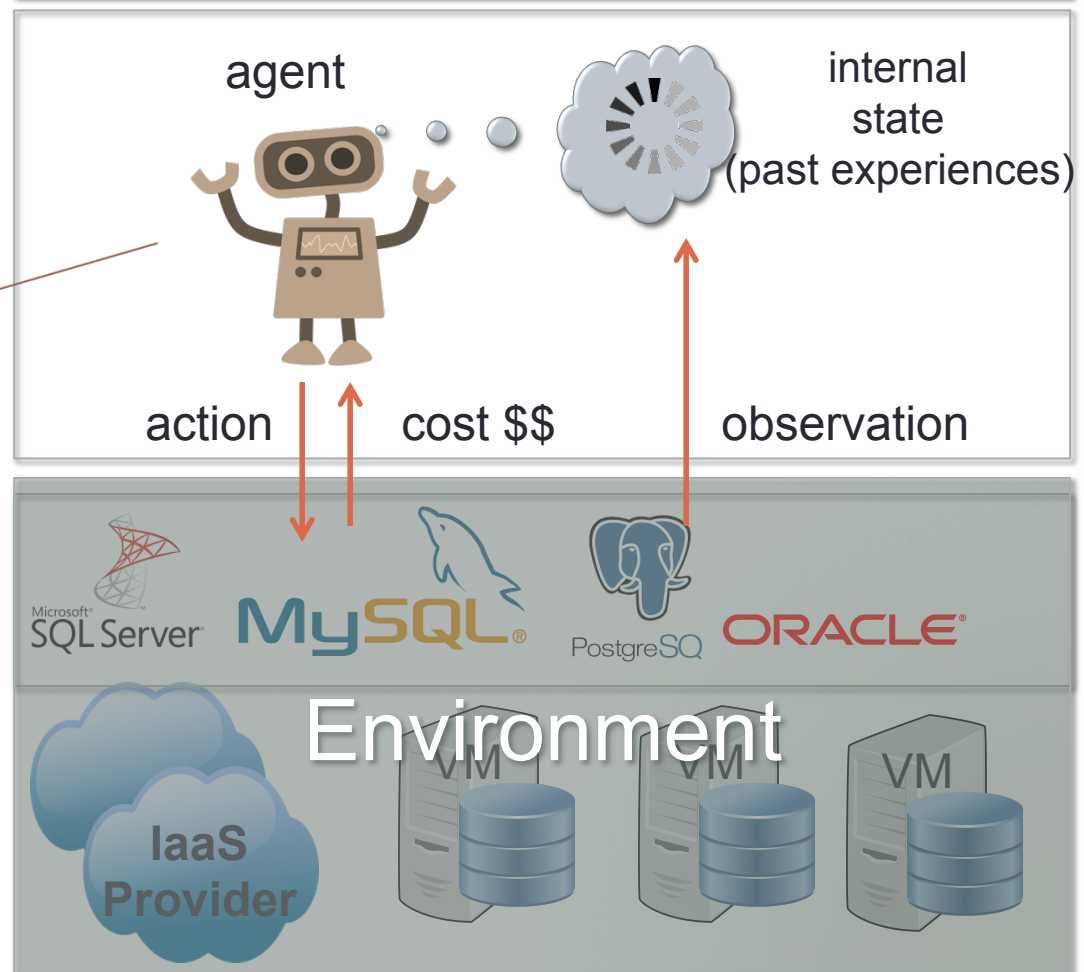


Data Management Application

Contextual Multi-Armed Bandit Problem

Slot Machine = Virtual Machine

*Which machine to use (new/old) (**action**) so that you execute the incoming query with minimum cost \$\$ (**cost**)?*



CMABs in WiSeDB

(Contextual Multi-Armed Bandits)



Action (per VM)

- Accept
- Pass to next /new VM
- Down one VM type

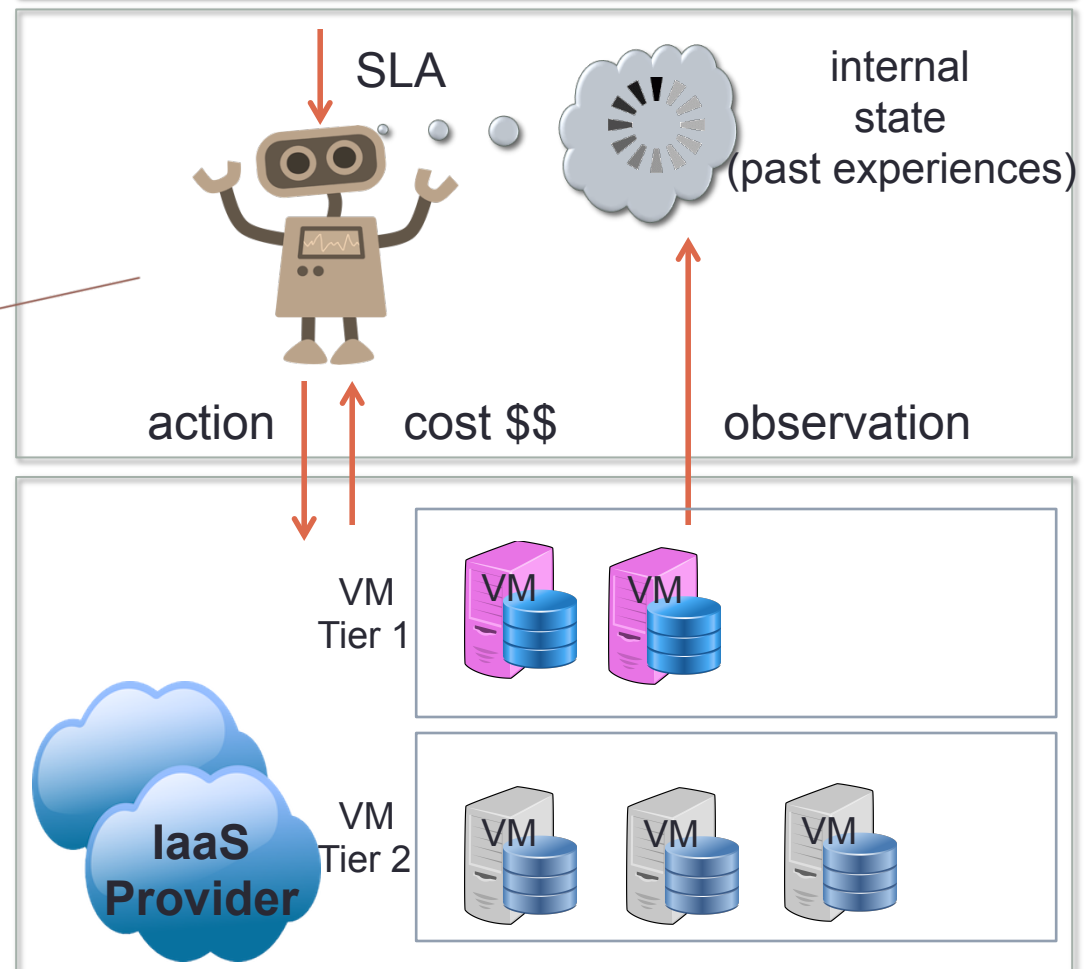
Reward

- \$\$ cost: processing & SLA violation penalties

Observation

- environment context (query, VM)
- action
- \$\$ cost

Data Management Application



CMABs in WiSeDB

(Contextual Multi-Armed Bandits)



Action (per VM)

- Accept
- Pass to next /new VM
- Down one VM type

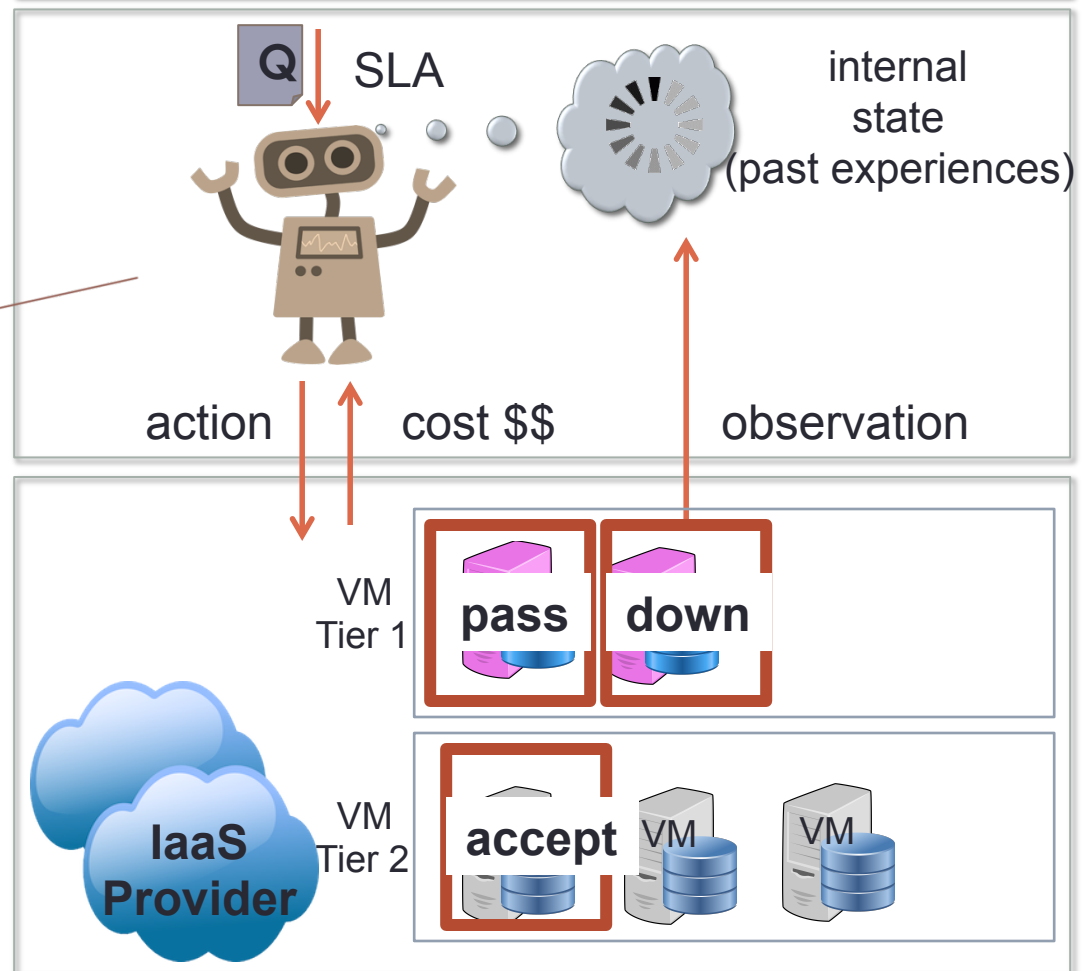
Reward

- \$\$ cost: processing & SLA violation penalties

Observation

- environment context (query, VM)
- action
- \$\$ cost

Data Management Application



CMABs in WiSeDB

(Contextual Multi-Armed Bandits)



Action (per VM)

- Accept
- Pass to next /new VM
- Down one VM type

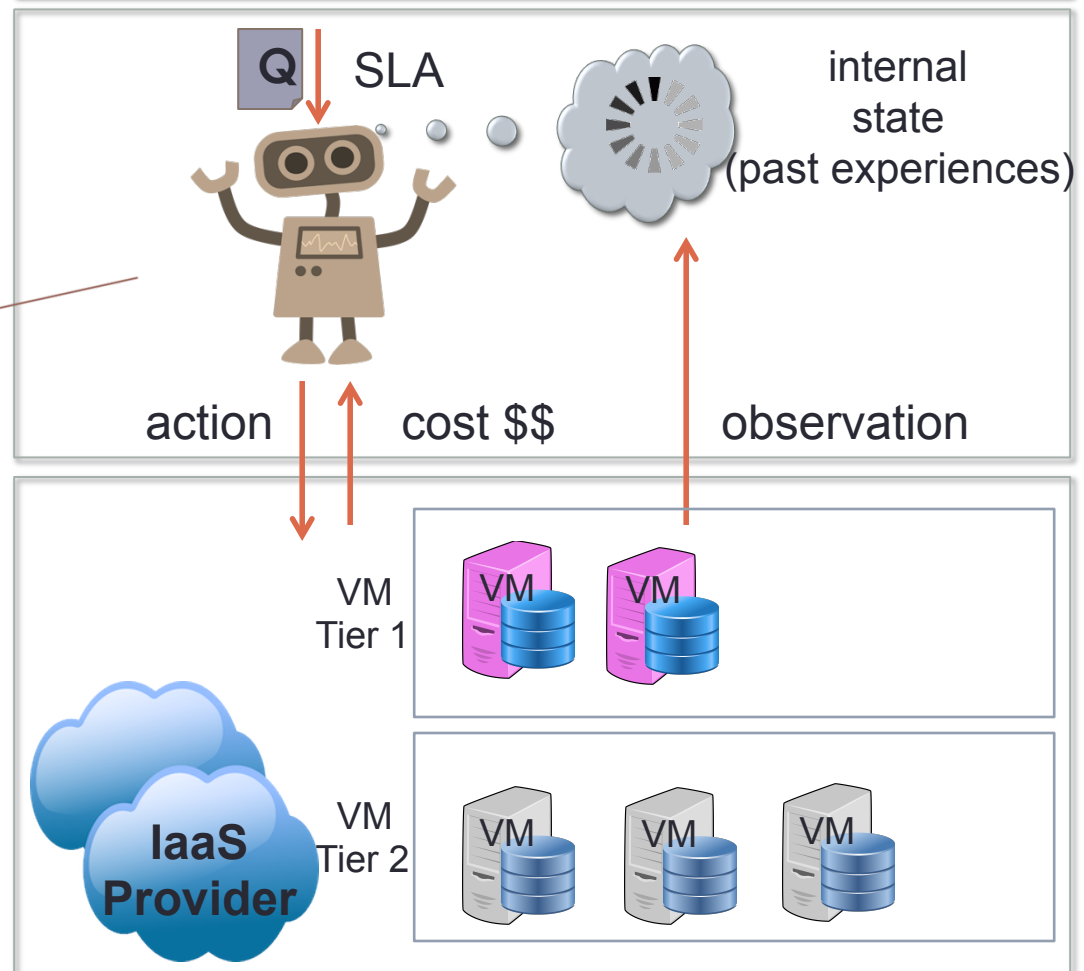
Reward

- \$\$ cost: processing & SLA violation penalties

Observation

- environment context (query, VM)
- action
- \$\$ cost

Data Management Application



CMABs in WiSeDB

(Contextual Multi-Armed Bandits)



Action (per VM)

- Accept
- Pass to next /new VM
- Down one VM type

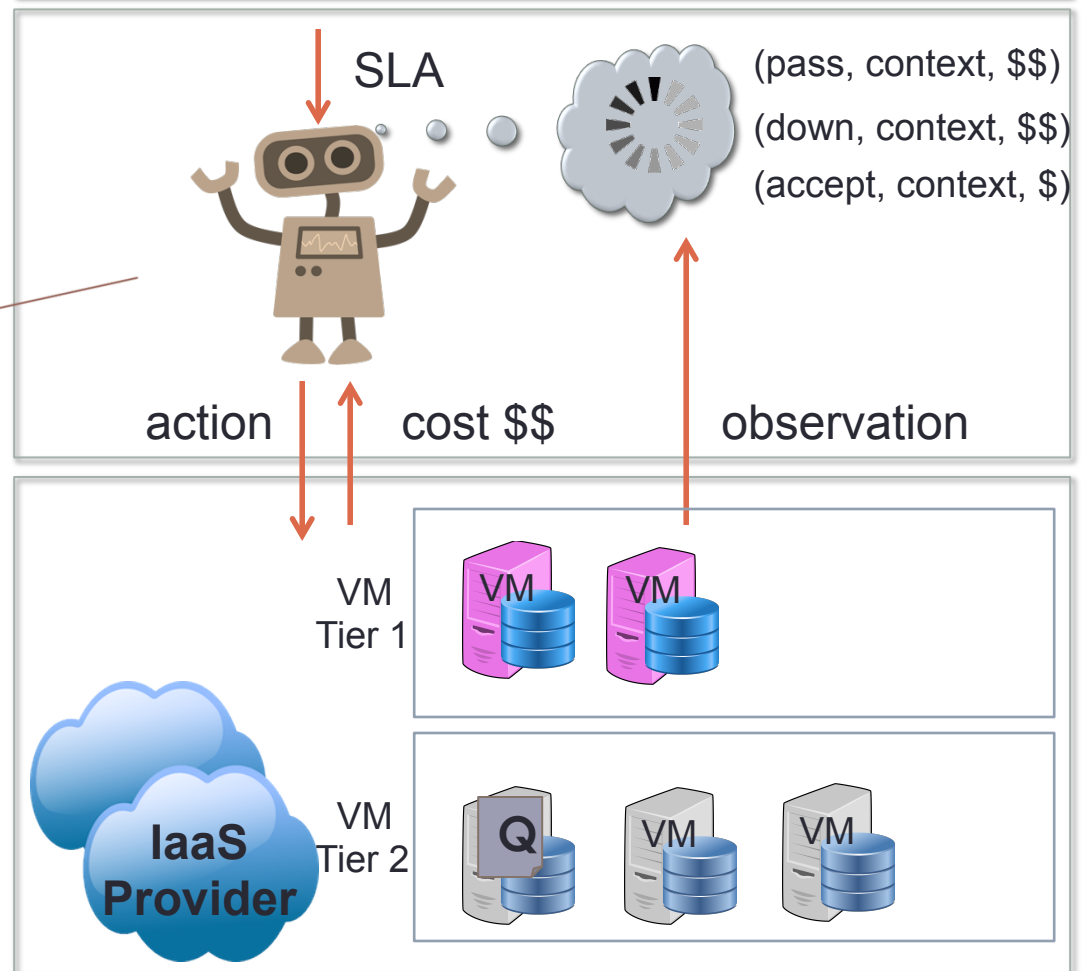
Reward

- \$\$ cost: processing & SLA violation penalties

Observation

- environment context (query, VM)
- action
- \$\$ cost

Data Management Application



CMABs in WiSeDB

(Contextual Multi-Armed Bandits)



Action (per VM)

- Accept
- Pass to next /new VM
- Down one VM type

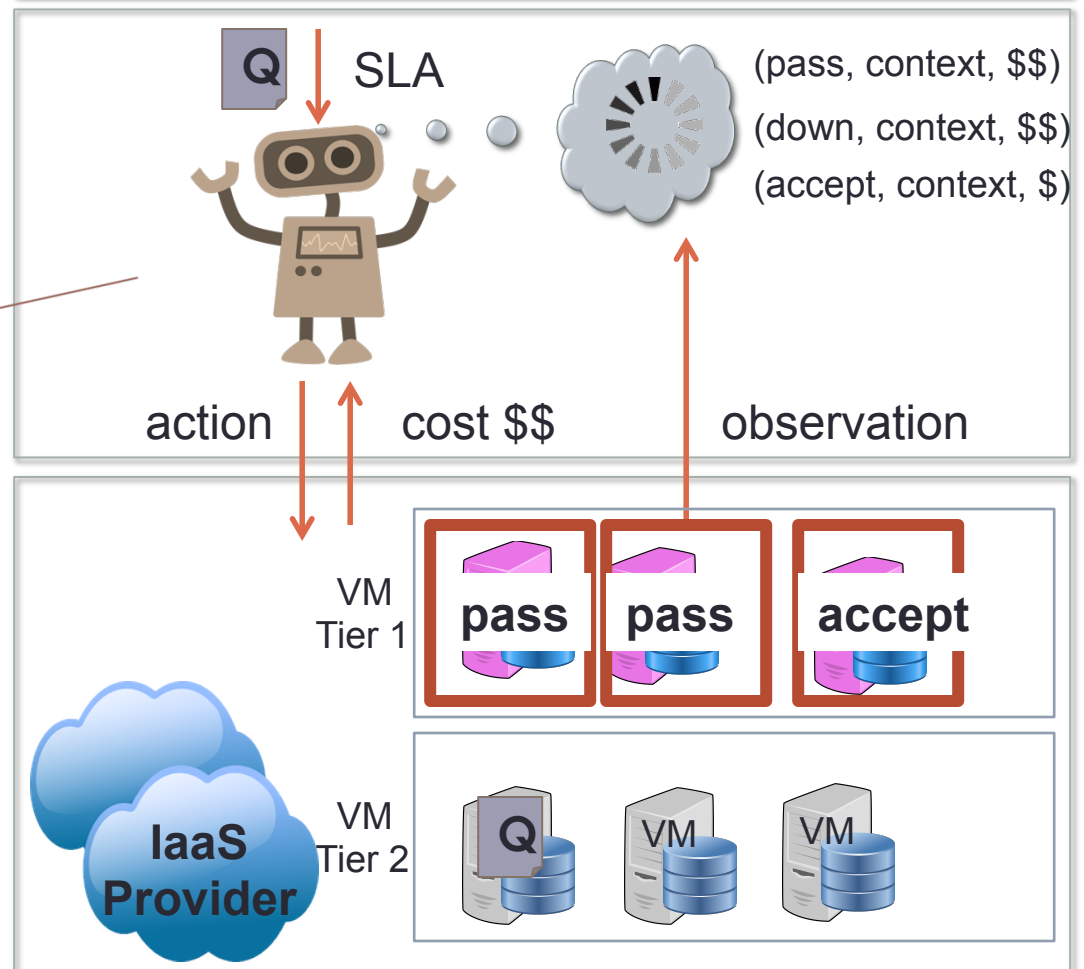
Reward

- \$\$ cost: processing & SLA violation penalties

Observation

- environment context (query, VM)
- action
- \$\$ cost

Data Management Application



Online Learning

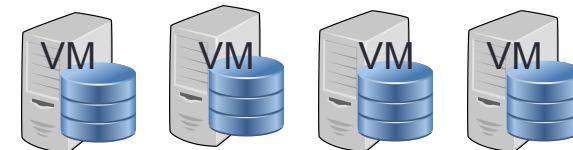
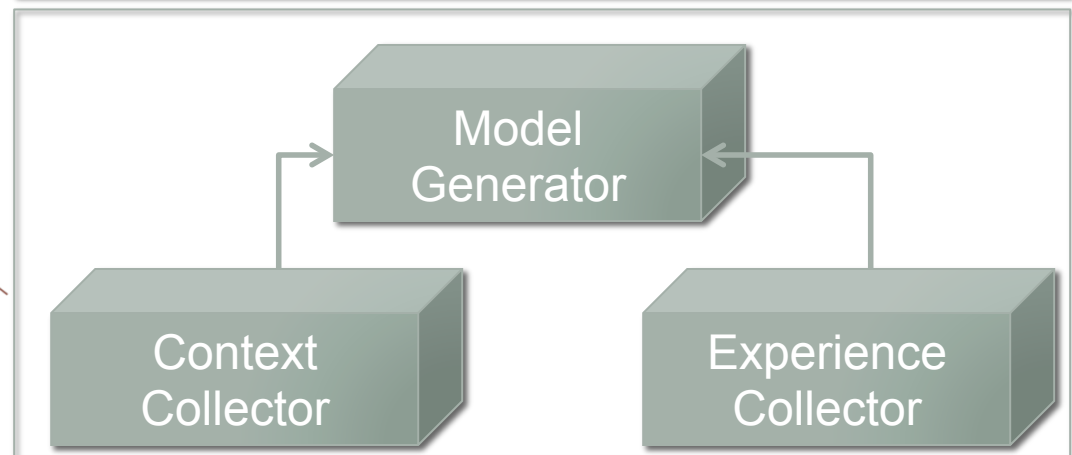


Context Features

- VM context**
 - memory, I/O rate
 - #queries in queue

- Query context**
 - tables used
 - # table scans
 - # joins
 - # spill joins
 - cache reads

Data Management Application



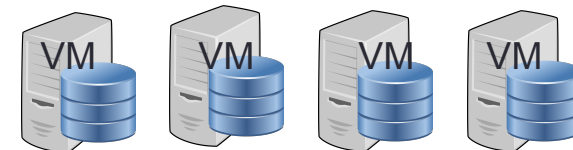
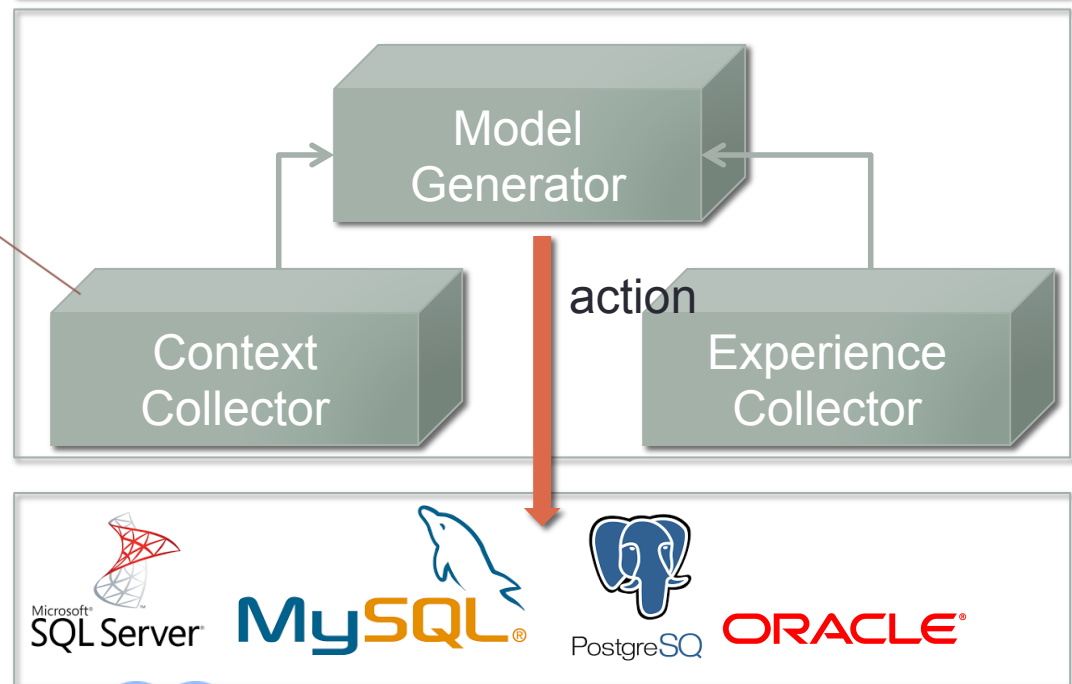
Online Learning



Action Selection

- Explore** opportunities
 - gather information
- Exploit** "safe" actions
 - make best decision given current information
- Thompson sampling

Data Management Application



Probabilistic Action Selection

- ❑ Select action according to probability of being the best

- ❑ Past observations $D = \{(x_i, a_i, c_i)\}$

- ❑ modeled by likelihood function over cost c : $P(c | \alpha, x, \theta)$

- ❑ **θ : parameters of likelihood function: splits of a regression tree**

- ❑ if (# joins in the query = 1) and (queries in the queue = 3) \Rightarrow cost = \$\$

- ❑ Posterior distribution of θ (Bayes rule)

$$P(\theta | D) \propto \prod P(c_i | a_i, x_i, \theta) P(\theta)$$

- ❑ $P(\theta)$: prior distribution of parameters θ

perfect decision tree is unknown



- ❑ Choose action α' to minimize cost for perfect model θ^*

$$\min_{\alpha'} E(c | \alpha', x, \theta^*)]$$

Probabilistic Action Selection

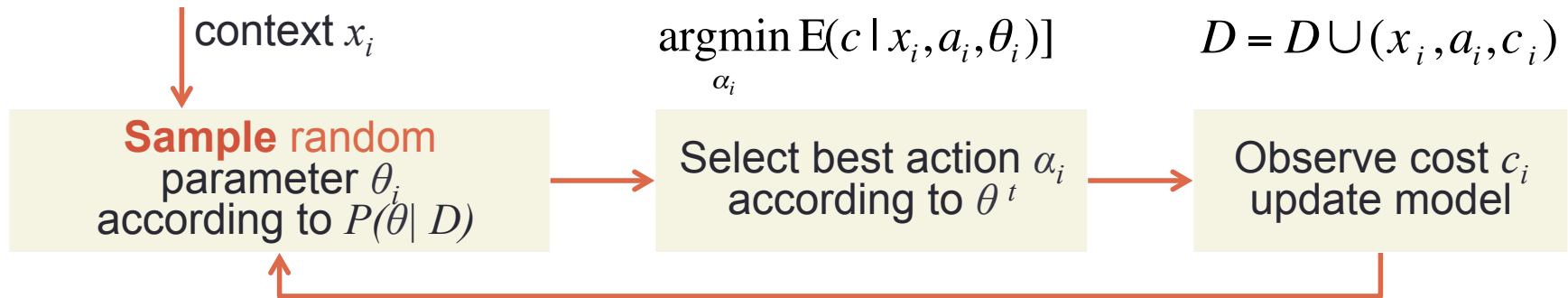
- ❑ Exploitation: pick action based on mean of posterior $P(\theta|D)$

$$\min_{a'} E(c | a', x) = \int E(c | a', x, \theta) P(\theta | D) d\theta$$

- ❑ Exploration: pick a random action
- ❑ Thompson Sampling: balance exploration/exploitation

Select random action according to probability that it is the best

WiSeDB Action Selection



**Select a random decision tree and
pick best action according to it**

Update the experience set

Create new model

Effectiveness

Training Data

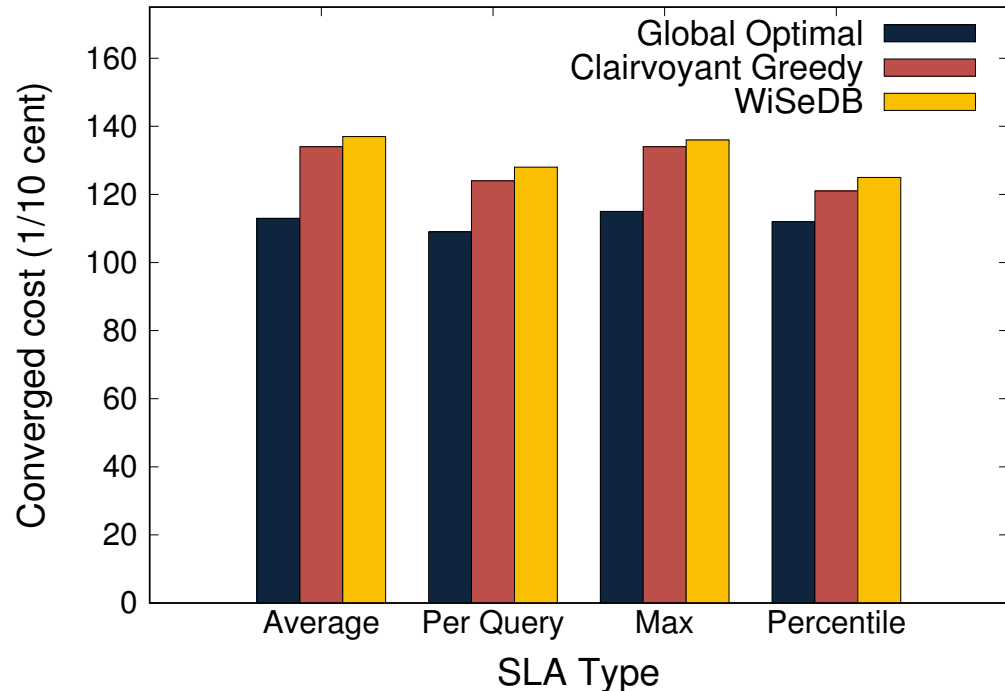
30 query sequence
22 TPC-H templates
repeat until convergence

Optimal: brute force (NP-hard)

Clairvoyant: perfect cost model

Amazon AWS

t2.large, t2.medium, t2.small



WiSeDB models can perform at the same cost as a perfect cost model

Effectiveness (concurrency)

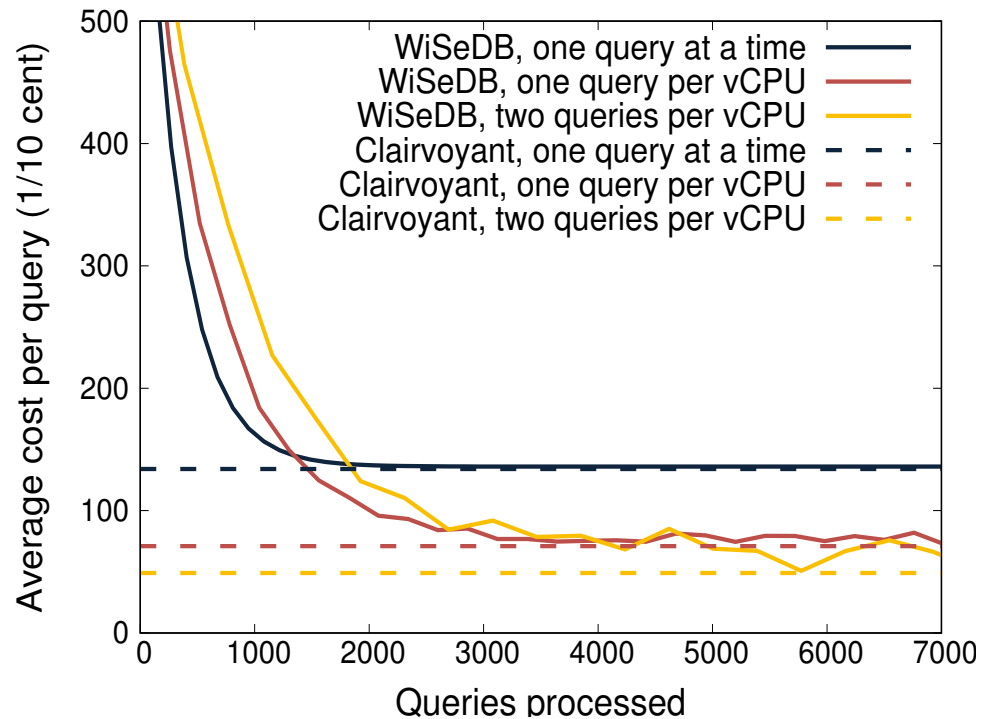
Training Data

22 TPC-H templates
900 queries/hour
Poison distribution

Clairvoyant: perfect cost model

One query/vCPU: 1-2 queries

Two queries/vCPU: 2-4 queries



WiSeDB models handles concurrency levels with no pre-training or tuning

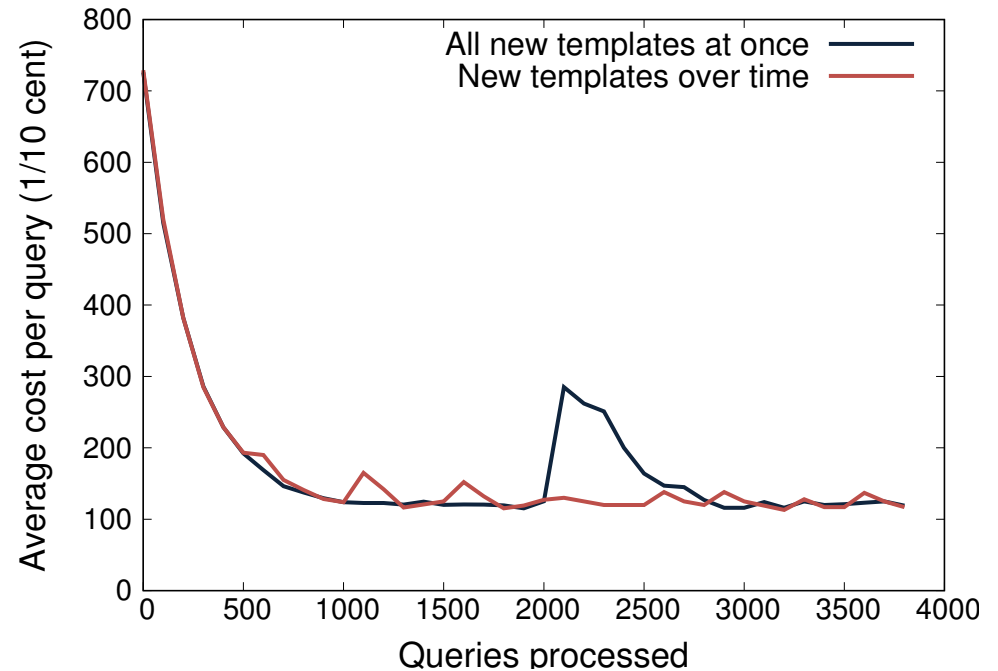
Adaptivity

Training Data

13 TPC-H templates
900 queries/hour
Poisson distribution
Max SLO

all new at once: 7 new templates
every 2000 queries (after
convergence)

new over time: 1 new template
every 500 queries



WiSeDB models quickly adapt to new unseen before templates

More details...

[VLDB 2016] *WiSeDB: A Learning-based Workload Management Advisor for Cloud Databases*, R. Marcus and O. Papaemmanouil (longer version on arXiv)

[CloudDB2016] *Workload Management for Cloud Databases via Machine Learning*, Ryan Marcus, Olga Papaemmanouil,

[CIDR 2015] *XCloud: Extensible Performance Management for Cloud Data Services*, Olga Papaemmanouil.

[EDBT 2014] *Contender: A Resource Modeling Approach for Concurrent Query Performance Prediction*, Jenny Duggan, Olga Papaemmanouil, Ugur Cetintemel, Eli Upfal

[CloudDB 2014] *SLA-driven Workload Management for Cloud Databases*, Dimokritos Stamatakis, Olga Papaemmanouil.

[DMC 2012] *Supporting Extensible Performance SLAs for Cloud Databases*, Olga Papaemmanouil.

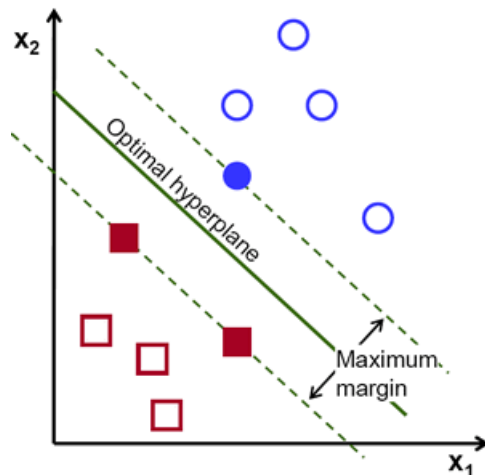
[SIGMOD 2011] *Performance Prediction for Concurrent Database Workloads*, Jennie Rogers, Ugur Cetintemel, Olga Papaemmanouil, Eli Upfal.

[SMDB 2010] *A Generic Auto-Provisioning Framework for Cloud Databases*, Jennie Rogers, Olga Papaemmanouil, Ugur Cetintemel.

Next Steps: Batch Scheduling



- Train once, use “**forever**”?
 - obsolescence detection and correction via SVMs



Data Management Application

Cost Management

SLA Management

Resource Provisioning

Workload Scheduling



ORACLE®

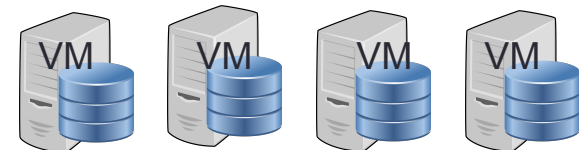
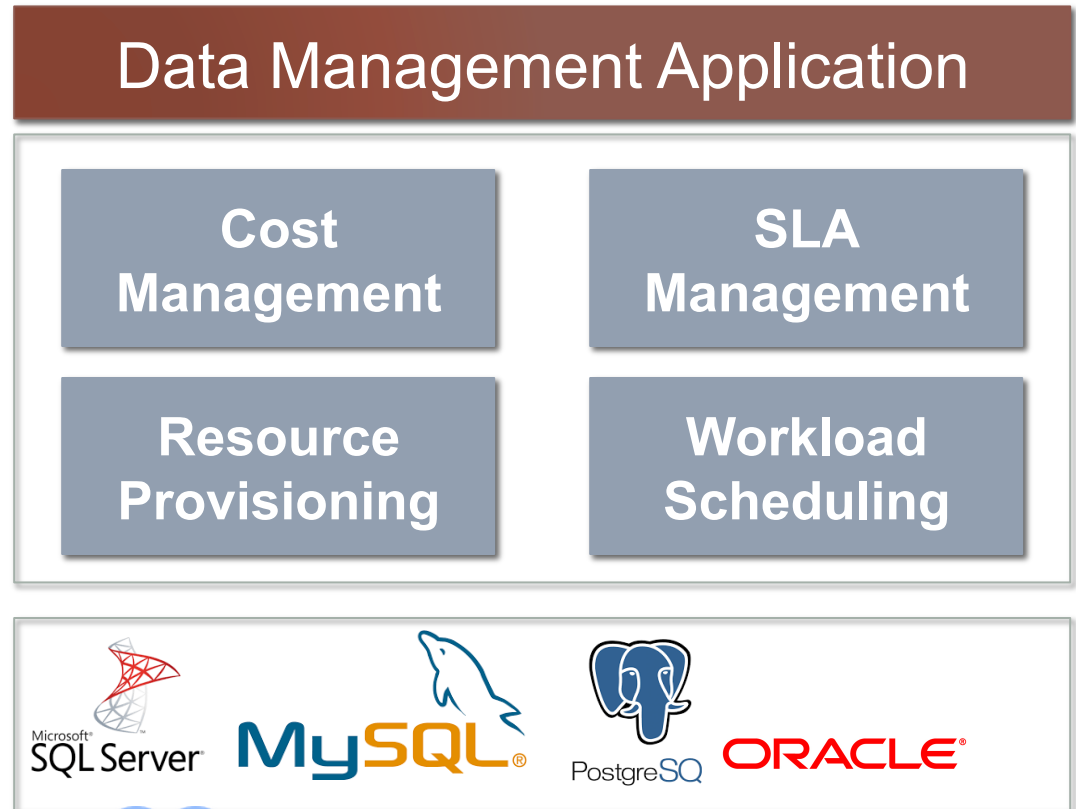


Next Steps



Batch Processing (Offline Learning)

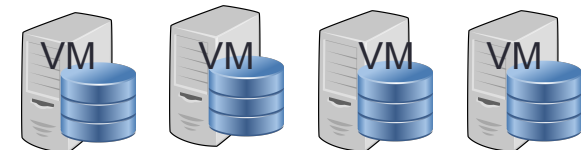
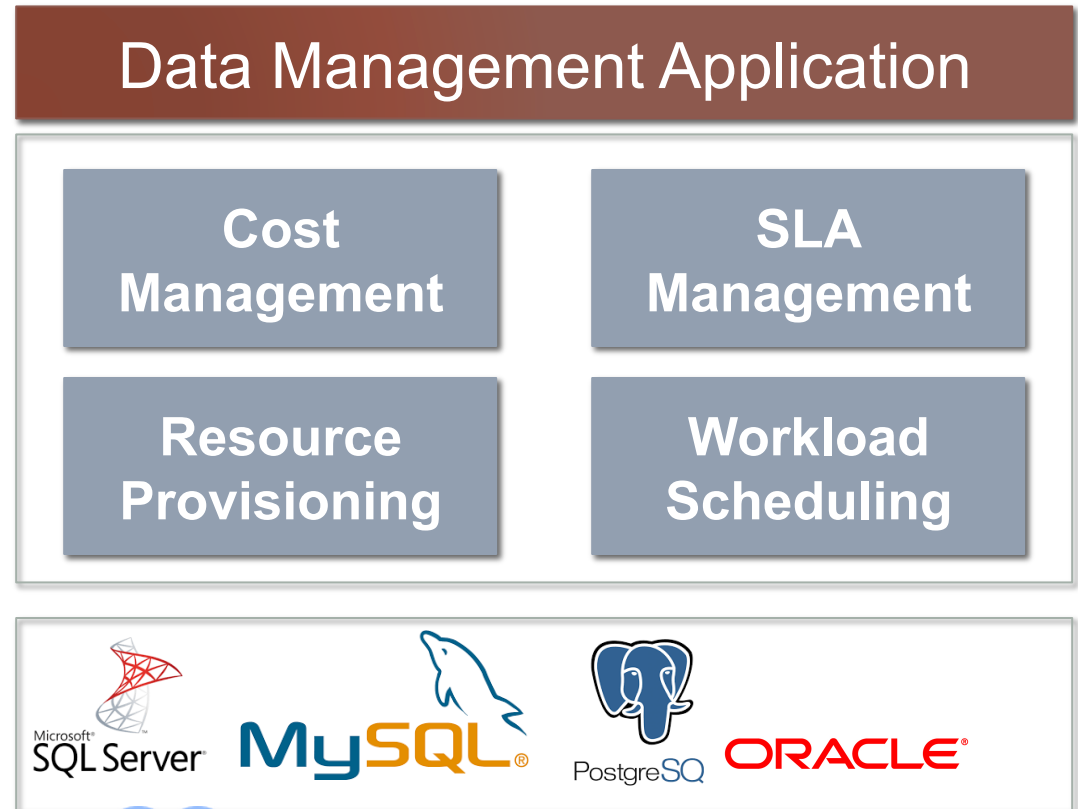
- ❑ Concurrent query execution
- ❑ Hybrid (offline/online) model
- ❑ Exploratory Query Execution



Next Steps: Online Learning



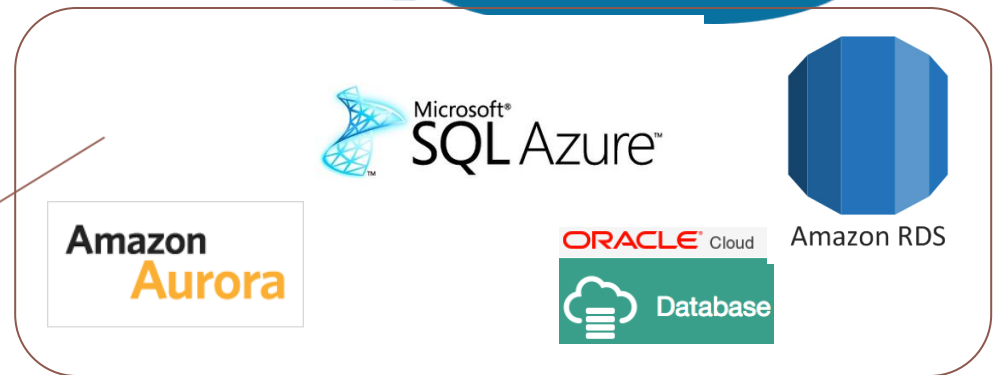
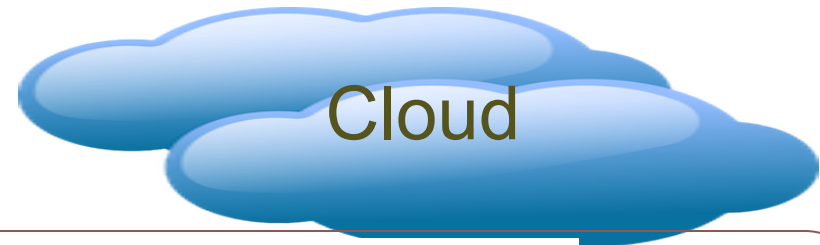
- ❑ Query Scheduling
 - ❑ query ordering actions
- ❑ Shut-down strategy
 - ❑ hill-climbing learning
- ❑ Training overhead
 - ❑ search space reduction



Next Steps: Tenant Placement

Database-as-a-Service

- Managed DBMS
- Relational & NoSQL DBs
- Cost effective tenants assignment to resources
 - SLO-awareness



Conclusions

- ❑ Cost and SLA management for IaaS-deployed DBs are not becoming simpler
- ❑ WiSeDB demonstrates how **ML techniques can help**
 - ❑ **discover** customized solutions for app-specific SLAs
 - ❑ **automate** complex application management decisions
 - ❑ **adapt** to workload and resource configurations
 - ❑ **build** systems that perform beyond unaided human heuristics

Our Database Group



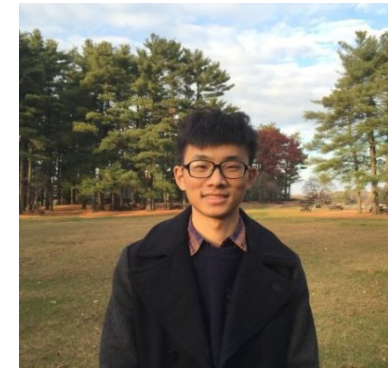
Ryan Marcus

Cloud Databases
Machine Learning



Kyriaki Dimitriadou

Interactive Data Exploration
Machine Learning



Zhan Li

Benchmarking Optimizers
Statistical Analysis

THANK YOU

Questions?